

27. Inventory Logistical Management (ILM)

27.1 XRP-II (Inventory Logistical Management {ILM})

ILM helps the M&O staff at the DAACs, EOC, and SMC to maintain records that describe all inventory components and their assembly structures and interdependencies. The database maintained by this tool, keeps chronological histories (a record of the transactions) of receipt, installation, and relocation of inventory items. ILM limits DAAC staff to accessing only those records, which correspond to equipment at their DAAC.

ILM is a set of automated tools that will assist the Procurement, Property Management, Maintenance, and Logistics teams in managing the tangible property of NASA's EOSDIS project. ILM is a heavily customized application developed utilizing the commercially available package XRP-II (Product Information and Inventory Management Modules). XRP-II is a manufacturing management system and its customization supports the required capabilities and functions of ILS. The application contains other functions in addition to the ILM tools. The ECS Management System Main Menu has selections for the Baseline Management tools which are not part of ILM.

XRP-II is a legacy based application. The application consists of a hierarchical menu screen structure and an imbedded COTS database (UNIFY). The hierarchical menu structure is built upon character oriented. **Once selected, entered, or deleted, the actions cannot be reversed.** In addition XRP/ILM is case sensitive. The vendor has customized the original screens to be ECS specific. The menu screens must be navigated to reach the appropriate ILM function. The lowest level of the structure is a function data screen that displays data and/or accepts data input for the function selected through the menu navigation process. While an operator is logged into the XRP-II program, he/she is engaged in a database session.

Use of the ILM tool involves considerations of the ECS business rules and general logistics concepts as applied on the ECS project. Some basic logistics definitions are provided below.

Each inventory item is identified by a unique Equipment Inventory Number (EIN), and in case of hardware items a physical silver sticker with the EIN is placed on the item. The most significant relationship maintained among inventory items is product structure. Product structure is the XRP-II term for the parent-component pairings that define the ingredients – or bill of material -- for an assembly. Product structures have corresponding active and inactive dates that establish the timeframe during which the pairing is in effect. They also facilitate tracking control item changes by a related configuration change request and/or trouble ticket.

27.2 ILM Operator Functions

The sections that follow present how to use the customized features of ILM. Each user/operator is assigned to a work group and the ILM menu options available are controlled based on the individual's role. Note: roles can be added or changed according to user need. The following roles currently exist within ILM:

- ILMADMIN All functions within the ILM;
- ILMADMND All functions require by DAAC System Administrator;
- ILMUSER All ILM operator privileges only;
- ILMLOG ILM Logistics User;
- ILMMAINT ILM Maintenance User;
- ILMMNTD ILM DAAC Maintenance User;
- LICUSER Software licensing privileges only;
- ILMQUERY ILM User with query privileges only;
- ILMUPDT ILM User with update privileges only;

Additionally there is an XRP administrator (XRPADM) who will have all privileges and is responsible for the operation of the XRP application. Customization of individual operator privileges is done on an as needed basis by an ILM system Administrator. The system tools provide the functions to revise the user privileges.

27.3 General Information

27.3.1 Using XRP

- The XRP application is case sensitive. It interprets data exactly as it is entered and takes the case of your input string into account. If something is in UPPER CASE, follow convention and put the request or data entry into UPPER CASE too.
- Pressing the <ENTER> key after each entry is required. Otherwise, the data entered may not be processed.
- The XRP application user interface is character based (not GUI). Keys handle navigation, selection, and moving. Each user interface screen has a set of active bottom line commands defining the keyboard letters, or function keys for activating functions or commands.
- **The mouse has limited capability within ILM.** ILM is not GUI driven; consequently, there is no cutting or pasting, placing the mouse pointer on an item and double clicking, etc. ILM is a character base system that requires the user to enter information, use bottom line commands, and press keys to start functions or commands.

27.3.2 ILM System

- The ILM System was designed to assist in the tracking of Government Property items for each site and in a consolidated manner.
- The ILM System is a character based, menu driven system based upon the UNIFY database.
- Each screen provides the user with simple and quick one or two keystroke commands to control entry and editing of data.
- System administrators have the ability to easily modify screens, menus, and reports to meet changing requirements and individual user needs.
- Included are functions for transferring data between sites and the SMC.
- Reports and screens at the SMC can provide consolidated views of material and requirements.

27.4 Quick Start Using ILM

ILM inherited a character-based user interface from the XRP-II application. ILM employs screens for data entry and report generation and menus for navigating to the screens. Data is entered via the keyboard in fields that are traversed from left to right, row by row. On data entry screens, labels for fields whose values can be modified are displayed in upper case; those that can not be modified have only the first letter capitalized. The database is updated every time a field's value changes, and a record of that change is written to a transaction log.

Most data entry screens have a form and a table view. Form views offer full screen layouts of a data record's fields, whereas table views offer rows of records in a window that is panned to see columns of fields. Some screens' table views, however, contain fewer fields than their corresponding form views. This is caused by system limitations on a table view's panes.

Numerous functions can be performed on the data entry screens. Commands available to an operator are screen-dependent and are listed near the bottom of each screen (hence their name: bottom-line commands). The **more** command helps the operator cycle through them. The terminology used can be confusing. "Mode" is used in two different ways: 1) as used in the next paragraph to describe data impact (Add, Insert, or Modify) and 2) the F4-mode of keyboard impact on the selected field in the display.

It is important to note that the UNIFY database management system XRP-II uses does not support rules requiring entries in specific fields. ILM attempts some enforcement via the data entry screens, either by establishing default values where feasible when new records are created, or by blocking an operator from advancing the cursor past a null field when in Add, Insert, or Modify modes. However, database updates can occur in ways that bypass these mechanisms, so operators must ensure required data is entered.

27.4.1 Invoking ILM from the Command Line Interface

To invoke the ILM program the operator must be logged in to the appropriate server and be registered with XRP for the appropriate privileges.

To execute ILM from the command line prompt use:

ilmusr

This script solicits the identity of the operator's workstation. When prompted the operator must provide either the workstation name or its IP address. This information is normally posted on a sticker on the workstation monitor. The script then starts XRP-II, and passes to the operator's userid it obtained from the system. The screen that is displayed after the XRP-II login is determined by XRP-II based on the user's ID and password.

Upon entering a valid userid and password the initial screen for the user will be displayed. The initial screen is associated with a specific userid/password. ILM users will be assigned to Groups according to the role for which their userid/password is valid. ILM privileges are dependent on the Group assignment. ILM privileges include function selections, data modification capability and report selection. *The configuration of the initial screen, screen modes, and function selection may result in the display of a data screen that is not exactly as shown in the presentation below.*

All ILM menus are similar in appearance and function the same way. Only the titles and selections vary.

Table 27.4.1-1. Procedures to Log into ILM

PERFORM	ACTION
Log on to XRP server at local site	At the Unix prompt type 'telnet <XRP server at local site>'
Provide log in information	At the login prompt Enter login name and press 'enter' Enter password and press 'enter'
Invoke ILM and display on the screen	Type 'ilm' and press 'enter' Type 'ilmusr' and press 'enter' Enter hostname, or IP address of the machine you are working on, so XRP knows where to display the screen

Remember to press <ENTER> after each field.

27.5 Commands and Functionality

The bottom line commands for each menu screen function the same as described for the main menu. Each data screen also has bottom line commands. Generally, the bottom line commands that are invoked with the single letter in bold print. For commands with an "already used" first letter, a slash, "/", is used as part of the command an example this would be /s command. A period, ".", is used for the third occurrence of a leading letter.

Note that the bottom line commands appearing on any screen are dependent on the user's attributes. Not all the commands listed for a screen may be appropriate to specific users.

A set of "standard" bottom line commands occurs on nearly all the ILM data screens.

While entering data into ILM you may notice that /zoom appears at the bottom of the screen for a given field. This is an indication that a ZOOM list is available to assist with data entry selections.

Table 27.5-1 summarizes XRP-II's bottom line commands for ILM. Sections 27.5.1 through 27.5.21 give the detailed definition for each of the bottom line commands.

Table 27.5-1. XRP-II's Bottom Line Commands for ILM (1 of 3)

Commands	Description
Commands used with ILM menus	
<F1>	Displays a description for the highlighted option.
<F3>	Moves back to the previous menu.
<F5>	Selects the highlighted option.
<F8>	Exit XRP-II.
Commands used with ILM screens	
<F1>	Invokes online help and displays a submenu for identifying the help target. Same as the H elp command.
<F2>	Clears the value from the field.
<F3>	Exits the screen or function. Same as the Q uit command.
<F5>	Starts a sort, select, find, or ad hoc report function after entry of parameters is completed.
<F7>	Copies data into or from a block of text.
<F9>	Tags and returns a value when executing a zoom command.
.Adjust_qtys	Updates how many of a license entitlement's node and user rights-to-use are allocated and remaining.
.Cartons	Invokes the cartons page on the EIN Shipment screen so the number and sizes of cartons in a shipment can be recorded.
.Entitlements	Activates an items page for identifying the license entitlements associated with a specific software license.
.Licenses	Activates an items page for identifying the software licenses associated with a specific license entitlement.
.Process_Changes	Updates property records based on parameters specified in the line items for a Maintenance Work Order.
/Add	Invokes ADD mode so new records can be added (created) in the database. New records are placed after the current record.
/Copy	Copies values from "tagged" fields to corresponding fields in other records. If no values are tagged, copies -- to the field in which the cursor resides -- the value from the corresponding field in the preceding record.
/Delete	Deletes the displayed record from the database.
/Insert	Invokes ADD mode such that new records can be inserted (created) in the database. New records are placed ahead of the current record.
/Modify	Invokes MODIFY mode so an existing database record can be updated.
/Note	Enables free-form text to be associated with a data entry screen for a user.
/Report	Invokes ad hoc report processing for the set of records currently selected on a screen.
/Sort	Allows the current set of selected records to be sorted according to operator-specified sort criteria.

Table 27.5-1 XRP-II's Bottom Line Commands for ILM (2 of 3)

Commands	Description
/Zoom	Allows a set of records related to the current record to be displayed. Tagging any field in one of those records causes a value from that record to be returned and entered in the field at which the command was invoked.
Addr	Invokes the vendor address maintenance screen so multiple addresses for a vendor can be recorded.
Bom	Invokes a screen to display the Bill of Material (i.e., list of first-level components) for an item, if any.
Changes	Displays the record of changes logged for a purchase order line item.
Check	Validates certain data entered for a batch of inventory transactions prior to the transactions being processed.
Copy-bill	Adds to an EIN's Bill of Material (BOM) the BOM from another.
Copy-dates	Copies active and inactive dates -- defined for an EIN's structure in its structure manager record -- into the product structure records for the EIN's children.
Copyein	Creates a new item by copying all the fields except the EIN Number from another item.
Copypart	Creates a new item by copying all the fields except the EIN Number from another item.
Duplicate	Creates copies of a purchase order line item to support multiple deliveries on different dates.
Execute	Starts the processing of a major, supporting function attached to the screen.
Find	Locates and displays the first record having field values the operator specifies. Repeating the Find command without changing the search criteria locates the next record that qualifies.
Go	Locates and displays a record having a specified sequence number. The format is "ng", where <i>n</i> is the number.
Help	Invokes online help and displays a submenu for identifying the help target.
Items	Invokes the items page of a data entry screen so a set of related records can be attached to the current record. Examples of related records include the line items for a purchase order and the components of a parent EIN.
Items_Addl	Invokes the items page of the License Allocation data screen so allocated licenses can be mapped to backup/redundant servers without being included in computations of rights-to-use consumed.
Items_Allocation	Activates an items page that lists the host machines and sites to which a license has been allocated. The license's rights-to-use must have first been mapped to at least one entitlement before the license can be allocated.
Justify	Used with table view, places the column the cursor is in next to the column(s) of record key data at the left edge of the screen.
Left	Shifts the data window to the left for displays that cannot fit all fields in one window.
More	Displays more bottom-line commands. In general, XRP-II provides three menus of bottom-line commands for screens, since all commands available to a screen cannot fit on one line. This command cycles through these menus.
Next	Moves the display "forward" to the next record (in form view) or next page of records (in table view).

Table 27.5-1. XRP-II's Bottom Line Commands for ILM (3 of 3)

Commands	Description
Prior	Moves the display "back" to the prior record (in form view) or prior page of records (in table view).
Quit	Exits the current screen or function. (This command is not available when in ADD, INSERT, or MODIFY modes, as it would be mistaken for a character being entered in a field.)
Right	Shifts the data window to the right for displays that cannot fit all fields in one window.
Select	Invokes query-by-example record filtering and displays a submenu for specifying the criteria to be used. See <i>XRP-II System Reference Manual</i>
Tag	Identifies a specific record and field whose value is to be used when adding new records or copying data. Tagged values are highlighted on the screen.
Untag	Removes the "Tag" from all field on the screen.
View	Toggles between "form" or record display and "table" or list display.
Where	Invokes a screen to display the first-level parents or assemblies having the EIN-controlled item as a component.
Write	Saves the current record to a file designated by the operator.
Commands used in ADD, INSERT, and MODIFY modes	
<F1>	Invokes online help and displays a submenu for identifying the help target.
<F2>	Erases the character string in the field.
<F3>	Exits the mode.
<F4>	Switches among typeover, insert, and replace modes for data entry.
<F6>	Enters the default value for the field.
Commands used in DELETE mode	
H	Invokes online help and displays instructions on how to use the Delete command.
L	Invokes the line-by-line method for deleting records.
n	Specifies the number of records to delete starting with the current record.
Q	Exits the mode.
Commands used with online help	
C	Display help for bottom-line commands available to the screen. Commands are listed on the bottom-line menu, and the More command can be used to cycle through them. Type any highlighted keys to display the help text for those keys.
F	Display help for the screen field on which the cursor has landed.
Q	Exit online help.
S	Display help for the screen.

27.5.1 Cursor Motion

The four arrow keys (UP, DOWN, LEFT and RIGHT) are used to move the cursor to various fields of the screen. A number may be entered before depressing an arrow key in order to move to the cursor multiple fields at once. The current number that has been entered is displayed at the bottom left hand corner of the screen. If a DOWN arrow is entered when the cursor is at the bottom of a table view screen, then, assuming there is more data in the file, the screen is redrawn to display the data shifted a half page down. Similarly, the display may be shifted a half page

upwards by entering an UP arrow at the top of the screen. Cursor motion via the arrow keys is limited to the current page plus a half-page shift.

27.5.2 Next or Prior

The bottom line “**n**” (Next) and “**p**” (Prior), when in the form view (one record on the screen) these commands move the display to the next or prior record. When in the table view (multiple records per screen) these commands move to the next or prior page of records. A number may be entered prior to the command as in “**10n**” which advances the display 10 records when in the form view or ten pages when in the table view.

27.5.3 View Command

This command toggles the display between the form and table views of the data. The table view shows basic information for several records at once. The form view shows all of the fields of the current record. For records with too many fields to fit across the screen when in the table view, the Left, Right, and justify commands described later may be used to move the display to the left or right.

27.5.4 Find Command

This command is used to find a record based on data entered. Data may be entered at one or more fields of the display and a partial string may be entered. Datalook (is a utility that searches the database for specified information) to clear the screen and allows the user to enter data into one or more fields to find. Pressing “**F5**” begins the search.

27.5.5 Go Command

The Go command allows the user to go to the First record, Last record, or the specified record number. To use the Go command type ‘**g**’ for Go. A prompt will come up stating “GO: First, Last, or Record Number, or Quit?” Type in the first letter to specify where to go (i.e. enter ‘**f**’ for first record). The entry of a number followed by ‘**g**’ for Go causes the display to shift to the specified record number. For example:

0g or **g** or **1g** moves to the start of the file

10g moves the cursor to record 10

1000g moves the cursor to record 1000

27.5.6 Select Command

The bottom line ‘**s**’ (Select) command allows you to select a subset of all the records in order to view, edit, or report on them. Once a set has been selected, the select command may be used again to select a subset of the set.

27.5.7 Selection Criteria

Data may be entered at one or more fields in order to specify the records to be selected. The selection function may be initiated either from the table or form views, but sometimes it is necessary to first switch to the form view if there is insufficient room to enter the desired selection specifications when in the table view.

There are two basic kinds of selection capability:

- a. **Exact matching**, where the user types exactly what the selected records are to contain.
- b. **Inexact matching**, where special characters are entered which are expanded into patterns during the matching process. The inexact matching provisions described below also apply to range fields (i.e. Account Number or Range).

Some examples of inexact matches are numeric and date ranges (for example, numbers from 1 to 100, dates before 1/1/87, or dates from 3/1/87 to 4/1/87), or substring matching (all the strings that contain the name “Smith”).

27.5.7.1 Exact Matches

To specify an **Exact** match, simply fill in the field or fields on the screen with the exact data to select for.

27.5.7.2 Inexact matches on String Fields

To specify an inexact match on string fields, use the following special characters.

- ? The “**wild character**”. The question mark matches any single character. Thus to find all the Smith’s whether spelled “Smith” or “Smyth”, use the specification “Sm?th”
- * The “**wild string**”. The asterisk matches any string of characters of any length, including zero length strings (also called “**null strings**”). A * is automatically appended to the end of all string specifications.
- [...] The character class framed by the brackets matches any single character that is a member of the class. For example, [apq] matches any of the letters a, p or q. Ranges of characters may be specified by separating 2 characters by a dash (“-”). All upper case letters could be represented by the class [ABCDEFGHJKLMN-O-PQRSTUVWXYZ] or more conveniently as [A-Z]. All letters, upper and lower case together, can be represented as [a-z]. Other classes can be similarly constructed.

If the string field contains numbers (eg. H0002) and the user wishes to inexact match on a range of these numbers it is important to understand that ranges within strings behave differently than ranges within numeric fields. For example a good range for a numeric field could be 1-9999. This is defined by the system to be all numbers ≥ 1 and ≤ 9999 . However, the range of [H00001-H99999] for a string would be defined as all strings starting with the letter H, or the numbers 0 and 9, or characters falling in the range of 1-H. If the user wanted to select all fields of H00001 through H9999, he could use H[0-9][0-9][0-9][0-9] if space permits.

27.5.7.3 Inexact Matches on Numeric Fields

Inexact matches on numeric fields, including dates and times, can be constructed by the Following set of expressions.

- >f1** The “**greater than**” operation. All fields with values greater than the entered value will match.
- <f1** The “**less than**” operation. All fields with values less than the entered value will match.
- !f1** The logical “**not**” operation. All fields that do not match the entered value will match.
- F1-f2** The “**range**” operation. All field values that match the entered values, or are between the entered values will match. This is equivalent to $\geq f1$ AND $\leq f2$.
- !f1-f2** This expression matches all field values that are outside the range of entered values. This is equivalent to $< f1$ OR $> f2$.

Any number of fields on the screen can be filled in as described above. The result is to select from the records of the file those which match all of the entered values. Once a set of records has been selected in this manner, Datalook displays the selected records.

27.5.8 Sort Command

The bottom line “/S” (Sort) command allows the current records to be sorted via any field or fields of the screen. Enter a number (1, 2,...) at each of the fields to be included in the sort, in the order they are to be included. Then press “F5” command to initiate the sort.

If a negative number is entered at a field, that field is sorted in descending order instead of the default ascending order.

27.5.9 Note Command

This command activates a text area for the user to write notes about the displayed screen. The information is retrieved by the same user when in the same screen.

27.5.10 Add/Insert

These commands are used to add new records. The bottom line “/I” (Insert) command adds a new record or records before the current record while the bottom line “/a” (Add) command adds a new record or records after the current record.

27.5.11 Modify Command

The bottom line “/m” (Modify) command is used to modify one or more fields of existing records. After modifying each field the cursor moves to the next field in the default direction, down in form view, or right in table view. Press “F3” to exit modify mode.

27.5.12 Delete Command

The bottom line “/d” (Delete) command offers a choice of deleting “line-by-line” or multiple records at once. If the “line-by-line” option is selected each entry of a down arrow deletes the current record. If, instead, a number is entered, the logic deletes the specified number of records.

27.5.13 Write, Execute Command

These commands appear only on screens that can drive an executable function. The bottom line “w” (Write) command is used to save the current record of the screen in a file named by the user. This file may then be referenced in a UNIX script that executes the function in **BATCH** mode. The bottom line “e” (Execute) command is used once the screen data is edited in order to start execution of the underlying function.

27.5.14 Items Command

The bottom line “**I**” (Items) command appears only on header-line item combination screens, such as are used for Purchase order or Work Order. When on the EIN menu (header record) pressing “**I**” activates the item screen.

27.5.15 Help Command

The bottom line “**h**” (Help) command on the main menu provides information concerning the screen, the fields of the screen, or Datalook commands. If the commands option is selected you may enter the letter or prefix plus letter that activates the command to see the help information for that command.

27.5.16 More Command

The “**m**” (More) command cycles the bottom line prompt through all of the available menu choices.

27.5.17 Quit Command

The bottom line “**q**” (Quit) command exits the screen.

27.5.18 Zoom Command

When the cursor is at a field which is related to data in a different table of the database, the bottom line “**/z**” (Zoom) command appears at the right of the screen. If the command is executed, Datalook opens a window to a different screen which displays the related data. The standard commands (Find, Go, Next, Prior, or arrows) may be used on the data in the window. If the user tags the field (the **tag** command is described later) and exits the zoom screen, the tagged value is returned to the initial screen. The “**/z**” (Zoom) option also appears when it is applicable while adding records or modifying fields.

27.5.19 Left, Right, Justify Commands

They allow the data window to be shifted left or right for screens, which have, too many fields to be shown on one page. The bottom line “**j**” (Justify) command causes the page to start with the current field at the left.

27.5.20 Tag, Untag Command

Tagged fields are used to identify default field values to be used when adding records or copying data into one or more records. If the user enters the “**t**” (Tag) command when the cursor is at a specified field, the field is tagged. This highlighted field in a manner (such as reverse video, or half intensity, depending on how the screen has been interfaced to the UNIX operating system). Only one field in a specific column of fields may be tagged at a time. A tagged field may be untagged either by entering the “**t**” (Tag) command again when the cursor is at that field, or by tagging some other field in the same column, or by entering the “**u**” (Untag) command which untags tagged fields. A tagged field remains tagged whether or not it is on the current screen until it is untagged.

27.5.21 Report Command

The bottom line “/r” (Report) command has a series of submenus which identify the report, its parameters, and the destination of the report. Every database maintenance screen has three built-in reports:

- a. a “Table Report” which follows the format of a table view screen, but adds a report header and pagination
- b. a “Form Report” which provides a single-page report of the form view for the current record
- c. An “ASCII Report” which displays the data in ASCII form without headers and pagination, suitable for loading into a spreadsheet or transmitting to another computer.

27.5.21.1 Selecting Records for Printing

Before printing a Table Report or an ASCII Report, the user may select a subset of records via the Select function. If no preselection has been performed, Datalook displays the numbers of the first and last records in the file and allows the user to accept these as the range of records to be reported on, or to modify them in order to report on a subset of the records. The first page of a table report shows the criteria used to select the records.

27.5.21.2 Column Selection

On Entering the Report Command, the user has the option of specifying the columns and the order in which they should be printed in a Table Report. If you do not specify the columns, the report starts with the left-most field screen and includes as many columns to the right as specified. The left-most field of the report may be controlled via the Left, Right, and Justify commands.

27.5.21.3. Report Format

The default ASCII Report format consists of data in ASCII form with fields separated by pipe (|) symbols. To generate a formatted report on a subset of the records, execute the select function before executing report.

27.5.21.4. Report Destinations

Once a report has been specified, Datalook offers the user a choice of report destinations. These choices may be 1) screen, 2) file, or 3) local printer.

If the report is not being sent to the screen, the user can specify whether or not it is to be run in background.

Once a report and destination have been chosen, the logic verifies that the width of the report is supported by the destination. If necessary, it retrieves the control characters necessary to put the output device into a mode such that it can support the width. If the report is too wide for the maximum width supported by the output device, the user is cautioned and may alter the destination or choose to proceed regardless.

27.6 ECS Management System Main Menu

The XRP top-level menu is ECS Management System Main Menu, illustrated in Figure 27.6-1 and described in Table 27.6-1. The userid/password configured initial screen will generally be different for the operator. The ECS Management System Main Menu contains selections that are not ILM functions. Baseline Management is covered in a separate document.

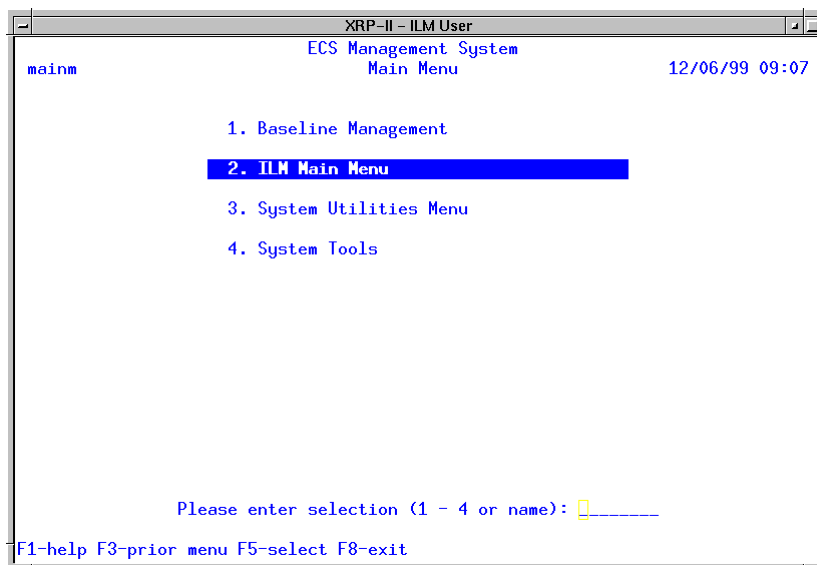


Figure 27.6-1. ECS Management System Main Menu

Table 27.6-1. Main Menu Functions

Menu item	Function	Section
Baseline Management	Provides access to XRP-II functions for maintaining control item and bill of material information	Refer to 609 section of BLM
ILM Main Menu	Provides access to XRP-II functions for maintaining inventory, logistics, and maintenance information	27.6.2
System Utilities Menu	Provides access to XRP-II functions for maintaining system information that spans functional domains	27.12
System Tools	Provides access to aids for registering XRP-II users, assigning permission, customizing data entry screens and menus, and performing general-purpose database dumps and loads.	27.13

27.6.1 ILM Main Menu

This menu (Figure 27.6.2-1 and Table 27.6.2-1) provides access to ILM functions for maintaining inventory, logistics, and maintenance information. The sections following focus on the functions and features that are part of the ILM main menu.

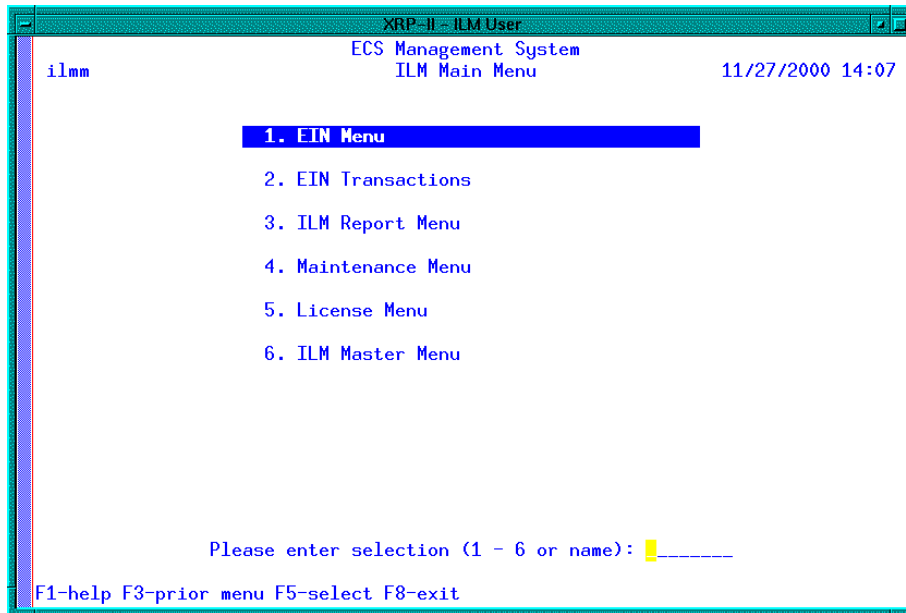


Figure 27.6.2-1. ECS Management System ILM Main Menu

The ILM Main Menu lets the operators navigate to the following submenus:

Table 27.6.2-1. ILM Main Menu Functions

Menu item	Function	Section
EIN Menu	For managing the catalog of EIN-controlled items	27.7
EIN Transactions	For browsing the log of all inventory transactions performed on items in the database	27.8
ILM Report Menu	For producing pre-defined reports	27.9
Maintenance Menu	For managing maintenance actions and data	27.10
License Menu	For managing licenses for commercial-off-the-shelf (COTS) software.	27.11
ILM Master Menu	For managing ILM parameters and reference information	27.12

27.7 EIN Menu

Options provided on the EIN Menu (see Figure 27.7-1 and Table 27.7-1) allow the operator to navigate to a set of screens for accessing the inventory information.

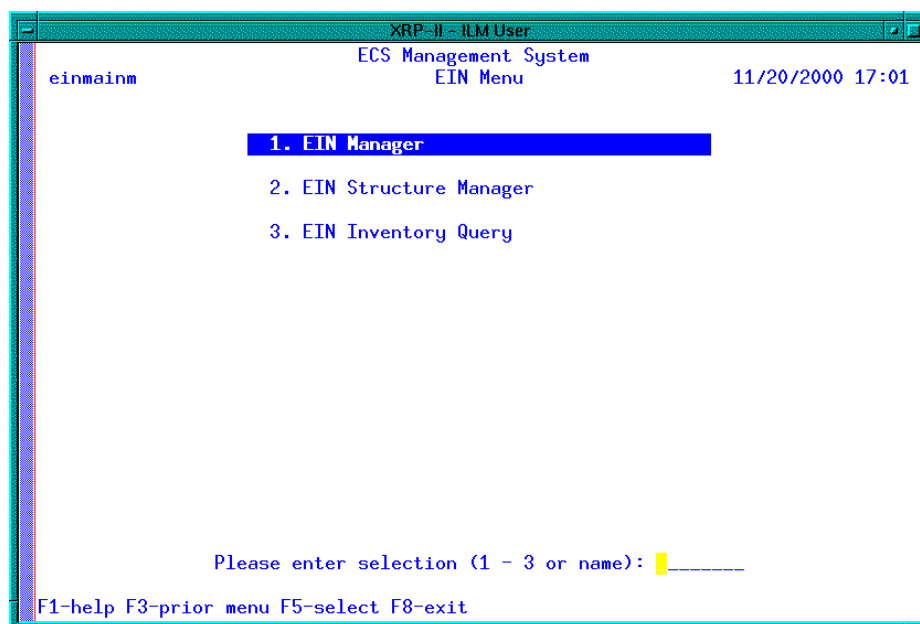


Figure 27.7-1. EIN Menu

The **EIN** menu is broken down into the following functions.

Table 27.7-1. EIN Menu Options

Menu item	Function	Section
EIN Manager	For browsing data describing EINs at the local site	27.7.1
EIN Structure Manager	For browsing EIN structures for items at the local site	27.7.2
EIN Inventory Query	For browsing EIN records	27.7.3

The following pages describe the screens, the data, and the process for reviewing EIN Controlled items data. Each selection item on the EIN menu is discussed, in the order on the menu screen.

27.7.1 EIN Manager

The EIN Manager screen (see Figure 27.7.1-1 and Table 27.7.1-1) is designed to view ILM EIN controlled items. This screen is always presented in the INQUIRY mode (see Table 27.7.1-2 for procedure). All changes to the database via this screen are reserved to the ILS Property Administrator and that function will not be active at the DAAC's. Operator may generate ad hoc reports on sorted or selected records. Refer to section 27.5.21 for instructions on how to generate ad hoc reports.

```

einmtd EIM Manager
LeinmtdJ EIN MANAGER: Last: 54547 Current: 2

EIN: 00000001
ECS NAME: n0mog10
SERIAL NUMBER: 35261778
OEM PART NUMBER: WB-R50XS-1G32
OEM DESCRIPTION: INDGO XS Graphics Workstation
HDWSFT CODE: H
MFG: SGI
MODEL/VERSION: XS
VENDOR: SGI
SOFTWARE LIC NUM:
MAINT VENDOR: SGI
WARRANTY EXP DATE: 10/31/02
NASA CONTRACT: NAS5-60000
PO Number: FFJ0000481
Tran Code: 03
LOCATION: NSIDC
ROOM: 231
Audit Date: 06/02/00
CONTROL ITEM ID:
Report Number: 5 G
COMMENT:
Code:
Note:

SUFFIX:
HOST ID:
Ecs Part Alias:
Resp Org:
YEAR MFG: 1993
RECEIVE DATE: 08/20/93
MAINT CONTRACT: CCH5697
STATUS CODE: I
RELEASE CODE:
INSTALLATION DATE: 10/06/93
BUILDING: NSIDC
USER:
GFE NUM: (NASA ECN)
RMA #:
Shipping Report Number: 0

Next Prior View Find Go Select Sort Note Copypart Bom Where Help More Quit

```

Figure 27.7.1-1. EIN MANAGER Screen

Table 27.7.1-1. EIN Manager Field Description (1 of 2)

Field Name	Data Type	Size	Description
EIN	String	20	Identifier for an EIN-controlled inventory item.
ECS NAME	String	30	Name of the machine with which the item is associated
SUFFIX	String	3	Code which when used as a suffix to ECS Name forms an identifier (RMA ID) for equipment subject to RMA reporting.
SERIAL NUMBER	String	30	Serial number of the item
HOSTID	String	20	Hexadecimal identifier of the host machine obtained when the "hostid" Unix shell command is run.
HDWSFT CODE	String	10	Code for classifying inventory items by type.
MODEL/VERSION	String	24	Model or version of the item
MFG	String	6	Code used for the manufacturer.
OEM PART NUMBER	String	34	Manufacturer or Vendor's part number.
OEM DESCRIPTION	String	40	Description of the OEM PART NUMBER entered in the field above.
ECS PART ALIAS	String	40	Common name used in ECS for a product and all its versions and variants.
RESP ORG	String	6	Code of the organization responsible for the item.
YEAR MFG	String	4	Year the item was manufactured.
RECEIVE DATE	Date	2	Date item was received from vendor
VENDOR	String	6	Vendor code whom the item was purchased from.

Table 27.7.1-1. EIN Manager Field Description (2 of 2)

Field Name	Data Type	Size	Description
SOFTWARE LIC NUM	String	10	License number for a software type license item.
MAINT VENDOR	String	6	Code for the vendor who is the maintenance vendor.
WARRANTY EXP DATE	Date	8	End date for the warranty period. This field default to 365 days from the date of entry.
MAINT CONTRACT	String	15	Maintenance Contract number for maintenance on this particular item.
STATUS CODE	String	1	Status of the item and is controlled by transactions within the system. The following codes are included : R - Received; S – Shipped; I - Installed; and X– Archived.
CONTROL ITEM ID	String	20	Identifier of the corresponding, version-controlled item in the Baseline Management system.
NASA CONTRACT	String	11	NASA contract number used for this item. The default number is NAS5-60000.
PO Number	String	10	Purchase order number against which the item was received.
Tran Code	Number	3	This field designates the transaction code. The value will always be set to '03'
Report Number	Number	4	Installation report number assigned by the system when an installation had occurred.
LOCATION	String	8	The actual location or site of where the item is.
BUILDING	String	6	Building number within the site where the item is.
ROOM	String	6	Room number where the item is or will be shipped to.
RMA#	String	6	Return Material Acquisition number.
RELEASE CODE	String	10	Code for distinguishing the release status of the item.
Installation Date	Date	2	The actual date this item was installed. The system sets the value during EIN installation processing.
Shipping Report Number	Number	2	Report number assigned to this item when the item was shipped.
USER	String	8	The user code of the person who has the item.
AUDIT DATE	Date	2	Date the item was physically inventoried last.
COMMENT	String	60	Miscellaneous information specific to the item.
NOTE	String	60	A message that can be associated with the item.

Table 27.7.1-2. Procedure to Inquiry for EIN

Perform	Action
Navigate to EIN Manager screen	From the Main Menu A. Select ILM Main Menu – press ‘enter’ B. Select EIN Menu – press ‘enter’ C. Select EIN Manager – press ‘enter’
To look for records of interest	A. Press ‘f’ to do a Find or ‘s’ to do a Select B. Enter the selection parameters (refer to section 27.5.6 for a more detail instruction on advance selection) C. Press ‘F5’ to start the search
To view the Bill of Material of the selected record	There is a command on the bottom of the screen call ‘Bom’ (Bill of Material). The Bom command lists the components of the selected item. If the selected item does not have any component attached to it, ILM will state that ‘No records available’. To invoke the Bom command, press ‘b’.
To find the parent EIN of the selected item	The ‘Where’ command at the bottom of the screen lists all the parents that the selected item has been assigned to and the associated active and inactive date.
Exit out of the current function or screen	Press ‘F3’ to exit the current mode.

Remember to press <ENTER> after each field.

27.7.2 EIN Structure Manager

The EIN Structure Manager screen (see Figure 27.7.2-1 and Table 27.7.1-1) is designed to display structure for a machine and items attached to it. This screen will only be presented to the user in INQUIRY mode. All changes to the database via this screen are reserved to the ILS Property Administrator and will not be active at the DAACs. Refer to Section 27.9.2 for instructions to generate EIN structure reports.

```

einstrcdEIN Structure Manager
Last: 2947 Current: 4

PARENT EIN: 00000004
Engineering Change: 00000000

OEM Part: PE301-CD
OEM Desc: 3000-300 Workstation
ECS Name: IN STOCK
Installation Report: 26 0
Ship Report: 0

PO NUMBER:
OR
VENDOR:

Date Entered: **/**/**
Operator Id:

ACTIVE DATE: **/**/**
INACTIVE DATE: **/**/**

Next Prior View Find Go Select /Sort /Note Copy dates Items Help More Quit Zoom
  
```

Figure 27.7.2-1. EIN Structure Manager Screen

Table 27.7.2-1. EIN Structure Manager Field Descriptions

Field Name	Data Type	Size	Description
PARENT EIN	String	20	Parent EIN for the installation/structure.
Engineering Change	String	8	Engineering change number assigned when the record was added to the database.
OEM Part	String	34	OEM part number of the parent EIN
OEM Desc	String	40	OEM Description of the Parent EIN.
ECS Name	String	30	Name of the machine with which the item is associated.
Installation Report	Number	4	Installation report number assigned by the system when an installation had occurred.
Ship Report	Number	3	Shipping report number assigned by the system when the item was shipped.
PO NUMBER	String	10	Purchase order number against which the parent EIN was received.
VENDOR	String	6	Vendor code from whom the item was purchased
Date Entered	Date	2	Date when this record was added to the database
Operator ID	String	8	Login ID of the user who added this item to the database and is not modifiable by the user.
ACTIVE DATE	Date	2	Date the item is received and entered into inventory.
INACTIVE DATE	Date	2	Date to make the structure ineffective.

27.7.2.2 Item Page of the Structure Manager

This screen (see Figure 27.7.2-2 and Table 27.7.2-2) is designed to view children items for the Parent EIN entered on the header page. This screen always comes up in Table view.

EIN CHILD	OEM PART	QTY PER	Last	Current
00007720	KEYBOARD	0.0000	2947	4
00007719	MOUSE - 3 Button	0.0000	2	1

Figure 27.7.2-2. EIN Structure Manager Items Page Screen

Table 27.7.2-2. Items Page Field Descriptions

Field Name	Data Type	Size	Description
EIN CHILD	String	20	Child EIN number of the item assigned to the Parent EIN.
OEM PART	String	34	OEM Part Number reflected from the EIN record of the child.
OEM DESC	String	40	OEM Description reflected from the EIN record of the child.
MOD/VER	String	24	Model/Version reflected from the EIN record of the child.
QTY PER	Number	3	This field is used more for consumable material application the parent items and reflects the quantity of the item that had been applied to the parent. This field defaults to quantity of one when the record is added. Database changes are reserved to the ILS PA
ACTIVE DATE	Date	2	Effective date on which the EIN child is assigned to the Parent EIN. NOTE: **/**/** = earliest system date.
INACTIVE DATE	Date	2	Effective date on which the EIN child is no longer assigned to the Parent EIN. NOTE: **/**/** = latest system date.

27.7.3 EIN Inventory Query Screen

The EIN Inventory Query screen (Figure 27.7.3-1 and Table 27.7.3-1) is designed to view the inventory location of EIN controlled items. The operator may sort and select by any field on the screen and then print a report of the data. This screen is displayed in INQUIRY mode only and the operator may not modify any data with this screen.

The screenshot shows a terminal window titled "einlocd EIN Inventory Query". The header bar displays "einlocd] EIN INVENTORY QUERY:" on the left and "Last: 54549 Current: 2" on the right. The main display area shows the following data for EIN 00000001:

```

EIN: 00000001
Location: NSIDC           Nat'l Snow and Ice
Building: NSIDC
Room: 231
ECS Name: n0mog10
Serial Number: 35261778
OEM Part: MB-R50XS-1G32
OEM Desc: INDGO XS Graphics Workstation
Mfr: SGI
Hdwsft Code: H
Control Item ID:
User:
Status: I
Install Date: 10/06/93
Audit Date: 06/02/00
  
```

At the bottom of the screen, a navigation bar contains the following options: Next, Prior, View, Find, Go, Select, /Sort, /Note, Help, More, and Quit.

Figure 27.7.3-1. EIN Inventory Query Screen

Table 27.7.3-1. EIN Inventory Query Field Descriptions

Field Name	Data Type	Size	Description
EIN	String	20	Identifier for an EIN-controlled inventory item.
Location	String	8	Identifier that designates an inventory location.
:	String	30	Description for the inventory location. It is obtained from the inventory location file based on the value in field Location.
Building	String	6	Identifier for the building where the item can be found.
Room	String	6	Identifier for the room where the item can be found.
ECS Name	String	30	Name of the machine with which the item is associated.
OEM Part	String	34	Manufacturer's or vendor's part number for the item.
OEM Desc	String	40	Manufacturer's or vendor's description for the item.
Mfr	String	6	Code used for the manufacturer.
Hdwsft Code	String	10	Code for classifying inventory items by type.
Control Item ID	String	20	Identifier of a corresponding version-controlled item in the BASELINE MANAGEMENT system.
User	String	8	Code of the person who has the item.
:	String	30	Name of the person who has the item. It is obtained from the employee file based on the value in field User.
Status	String	1	Code that designates the status of the item. The following values are set when processing transactions: R = Received; S = Shipped; I = Installed; X = Archived;
Install Date	Date	2	Date the item was installed.
Audit Date	Date	2	Date the item was physically inventoried last.

27.8 EIN Transactions

The EIN Transactions menu (Figure 27.8-1) provides access to a screen for browsing the log of past inventory transactions.

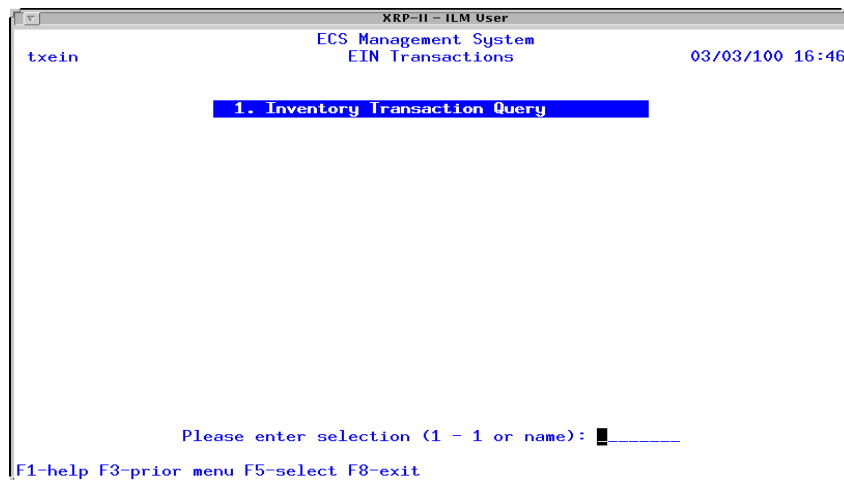


Figure 27.8-1. EIN Transactions

27.8.1 Inventory Transaction Query Screen

This screen (see Figure 27.8.1-1) allows operators to browse the log of all inventory transactions performed on items in the database. The operator may sort and select on any field on the screen and print ad hoc reports of sorted data, if desired, using XRP-II's report command. Table 27.8.1-1 describes the screen's fields.

The screenshot shows a terminal window titled 'Inventory Transactions Query'. It displays a form with various fields and their values. The fields are organized into two main columns. The left column contains: Site (ASF), Number (5632), Type of Transaction (SHP), Date (05/20/98), Time (14:13), EIn (C0000004), Transaction Id (0), Vendor Id, From Site (ASF), From Bldg, Room, Parent EIn, Order Type, Order Line No (0), Quantity (1.0), Operator Id (bcooper), and Reason Code. The right column contains: Dest Site (EDF), Dest Bldg, Room, Parent EIn (C0000004), Order Type, Order Line No (0), Quantity (1.0), and Operator Id (bcooper). At the bottom, there is a menu bar with options: Text, Prior, View, Find, Go, Select, Sort, Note, Help, More, and Quit.

Figure 27.8.1-1. Inventory Transaction Query Screen

Table 27.8.1-1. Inventory Transactions Query Field Descriptions (1 of 2)

Field Name	Data Type	Size	Description
Site	String	6	Code for the site that entered the transaction.
Number	Numeric	8	Record number of database record being observed.
Type of Transaction	String	3	Code assigned to the type of transaction being performed. INS – Installation; REL = Relocation; TR = Transfer; ARC = Archive; SHP = Shipment; RX = Receipt; FAI = Failed, MRV = Maintenance Replaced by vendor; REP = Replaced; and MTR = Maintenance Transfer.
Date	Date	2	Date the transaction was entered.
Time	Time	2	Time the transaction was entered.
Dest Site	String	6	Code for the inventory location gaining the item.
EIN	String	20	EIN of the item involved in the transaction.
Dest Bldg	String	6	Identifier for the building gaining the item.
Transaction Id	Numeric	6	Number assigned to a particular transaction
Room	String	6	Number for the room gaining the item.
Vendor Id	String	6	Code for the vendor from whom the item was purchased.
Parent EIN	String	20	EIN of the parent item involved in the transaction.
Order Type	String	2	Code for the type of order, if any, involved in the transaction.
From Site	String	6	Identifier for the building losing the item.

Table 27.8.1-1. Inventory Transactions Query Field Descriptions (2 of 2)

Field Name	Data Type	Size	Description
Order	String	6	Identifier for the order, if any, involved in the transaction.
From Bldg	String	6	Identifier for the building losing the item.
Line No.	Numeric	4	Line number of the item on the order if an order is associated with the transaction.
Room	String	6	Number of the room losing the item.
Parent EIN	String	20	EIN of the parent item losing the item.
Quantity	Floating	10.1	Number of items in the transaction.
CCR #	String	30	Identifier for the CCR authorizing the transaction.
Tt	String	15	Identifier for the trouble ticket associated with the transaction.
Operator Id	String	8	Login ID of the operator performing the transaction.
Reason Code	String	4	Code for the reason for the transaction.

27.9 ILM Report Menu

ILM Report Menu (see Figure 27.9-1 and Table 27.9-1) provides access to display and report controlled items in the database. This section of ILM is mainly used for reporting purposes. When the user authorization is more limited, this menu offers fewer options. The ILM Report menu lets the users navigate to the following screens:

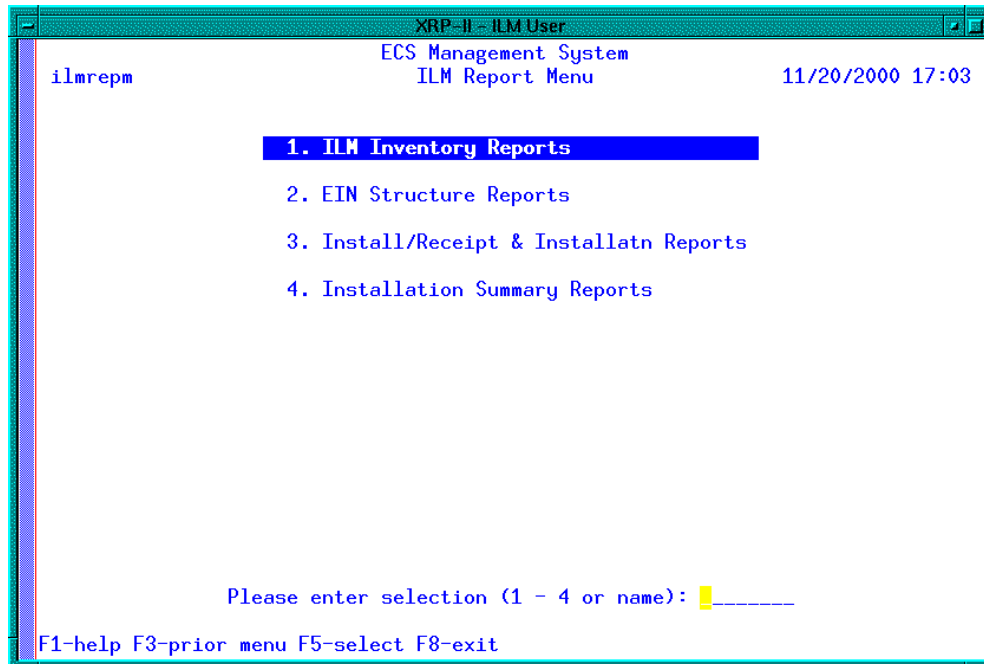


Figure 27.9-1. Report Menu

Table 27.9-1. ILM Reports Menu options

Menu item	Function	Section
ILM Inventory report	For printing all items contained within the designated location(s) by local site coordinators.	27.9.1
EIN Structure reports	For printing all designated parents and components in a multi-level bill report.	27.9.2
Install/Receipt Report	For printing a report of a parent EIN configuration and send the hard copy to the receiving organization for sign off.	27.9.3
Installation Summary Report	For printing a list of EINs installed during a specified time frame	27.9.4

27.9.1 ILM Inventory Reports Screen

The ILM Inventory Reports screen (Figure 27.9.1-1 and Table 27.9.1-1) is designed to retrieve and print all items contained within designated location(s). See Table 27.9.1-2 for procedure and Figure 27.9.1-2 for sample report.

ilminvd ILM Inventory Reports

ilminvd] ILM INVENTORY REPORTS: Last: 9 Current: 1

LOCATION: aRC [NULL for all]

NOTE 1:

NOTE 2:

NUMBER OF COPIES

INVENTORY REPORT - BY LOCATION: 1

Next Prior Find Go Select Sort Note Execute Help More Quit Zoom

Figure 27.9.1-1. ILM Inventory Report

Table 27.9.1-1. ILM Inventory Reports Field Descriptions

Field Name	Data Type	Size	Entry	Description
Location	String	8	optional	Code for an inventory location. The operator may zoom to the Inventory Location table and choose the code, if it had been entered there previously. (See the Inventory Location Manager screen.)
Note 1, 2	String	60	optional	A 60-character note to include in the report.
Inventory Report - By Location	Numeric	2	required	Number of copies of this report to generate.

Table 27.9.1-2. Procedures to Generate ILM Inventory Report

PERFORM	ACTION
Navigate to EIN Structure Reports screen	From the Main Menu A. Select ILM Main Menu – press 'enter' B. Select ILM Report Menu – press 'enter' C. Select ILM Inventory Report – press 'enter'
Invoke the add command	Press 'a' to go into add mode.
Enter parameters for the report	Fill in the necessary information A. Enter the location value or zoom to the inventory location table to pick the location to report. To do this, press ('/z', 't', 'F3') . B. Enter any note to appeal on the report header C. Enter number of copies of the report to generate
Exit the add mode	Press 'F3' to exit adds mode.
Execute the report	Press 'e' to execute the report.
Report output	View the report output options: A. Enter 1 and press 'enter' – to view the report on the screen OR B. Enter 2 and press 'enter' – to save the report into a file. Enter the file name to be saved as. The file will be saved in the user's home directory. OR C. Enter 3 and press 'enter' – to print the report to the defaulted printer
Exit the report menu	If option 1 was selected, go to the next step. If option 2 or 3 was selected, press 'F3' to exit the report menu when finished.
Navigate through the report output	A. Navigating through the report -Press 'n' for Next -Press 'p' for Previous -Press 'r' for Right -Press 'q' for Quit or -Press 'h' to print a hard copy of the report B. After finished making the selection, press 'q' to exit the report screen. A message will come up specifying the number of reports generated, press 'enter' . C. Another message will prompt "Another?" -Press 'y' to generate more reports. This will go back to the EIN Structure Reports screen, or -Press 'n' to go back to the ILM Report Menu.

Remember to press <ENTER> after each field.

(ilminv1)

ILM INVENTORY REPORT - BY LOCATION

DATE: 01/07/00 TIME: 12:36
PAGE: 1

LOCATION: EDF : ECS Development Facility

EIN	OEM PART NO	OEM DESC	MODEL	SERIAL NO	BUILDING	ROOM
00000000	PARENTREC	PARENT FOR NON INSTALLED ITEMS RM 1073			1616	1073
00000004	PE301-CD	3000-300 Workstation	300X AXP	AB3500171X	1616	1073
00000006	7012-340	RISC 6000 Workstation	6000	MS70122663304	1616	1073
00000007	A2094A	Color Monitor - 19 IN		JP01000992	1616	1100D3
00000008	VRT19-HA	Color Monitor - 19 IN		IS33984574	1616	1073
00000009	7208-001	4 Milimeter Tape Unit	Model 7208	MS72062626430	1616	1073
00000010	6091-191	19 Inch Color Monitor		23-K0146	1616	1073
00000011	A2627A	715-50 PA RISC Workstation	715-50	6342A30521	1616	1100D3
00000013	S10TX-44-032-P46	SPARCStation 10	10	403F1014	1616	3039
00000014	A2094A	Color Monitor - 19 IN		JP04050797	1616	1100D3
00000015	X557A	CD ROM - 644 MB		405G1578	1616	1100D7
00000016	TLZ06-VA	Tape Drive - 4 MM		CX35103575	1616	1073
00000018	X814A	Tape Drive - 5 GB - 8 MM		407G3165	1616	1100D4
00000019	C1521B	Tape Drive - 2.0 GB - 4 MM		3314E62862	1616	1052C
00000022	PE301-CD	3000-300 Workstation	300X AXP	AB333001N2	1616	1105B1
00000023	PE301-CD	3000-300 Workstation	300X AXP	AB33300I04	1616	1073
00000025	VRT19-HA	Color Monitor - 19 IN		IS31773470	1616	1073
00000027	VRT19-HA	Color Monitor - 19 IN		IS31162480	1616	1105B1
00000028	VRT19-HA	Color Monitor - 19 IN		IS31162482	1616	1073
00000030	X545A	1.05 GB HD - Desktop		410G0301	1616	1100F4
00000031	BA353-AF	CD ROM - in Storage Expansion Unit		KB34203698	1616	1073
00000033	X545A	1.05 GB HD - Desktop		412G2197	1616	1073
00000034	PE301-CD	3000-300 Workstation	300X AXP	AB3500305S	1616	1073
00000035	X557A	CD ROM - 644 MB		408G0598	1616	1100D4
00000038	4-30-GX-32 P46	SPARCSystem LX Workstation	LX	411E0158	1616	1073
00000040	A2627A	715-50 PA RISC Workstation	715-50	6342A30520	1616	1105A2
00000041	A2608A	735 CRX Performance Workstation-Server 3	735-CRX	6342A00425	1616	1100D3
00000042	A2627A	715-50 PA RISC Workstation	715-50	6342A30034	1616	1073
00000043	A2627A	715-50 PA RISC Workstation	715-50	6340A30125	1616	1073

Figure 27.9.1-2. ILM Inventory Report – by Location

27.9.2 EIN Structure Reports

This screen (see Figure 27.9.1-1) is designed to retrieve and print all designated parents and components in a multi-level bill report. Table 27.9.2-1 describes the screen's fields. Table 27.9.2-2 provides instruction how to generate the report. Refer to Figure 27.9.2-2 for a sample report.

The screenshot shows a window titled "einstrepEIN Structure Reports". Inside, there are several input fields with labels and values:

- EIN OR RANGE:** 00001029
- NUMBER OF LEVELS TO EXPLODE:** 99
- EXPLOSION QUANTITY:** 1
- DATE OF BILL:** **/**/** [Enter 1/1/1 for current date]
- TYPE OF SORT (SN):** [null=part S=sort string N=sort #]
- NOTE 1:**
- NOTE 2:**
- ENTER NUMBER OF COPIES**
- EIN MULTI-LEVEL REPORT:** 4

At the top right, it says "Last: 3" and "Current: 1". At the bottom, there is a menu bar with options: Next, Prior, Find, Go, Select, Sort, Note, Write, Execute, Help, More, Quit, Zoom.

Figure 27.9.2-1. EIN Structure Reports Screen

Table 27.9.2-1. EIN Structure Reports Field Descriptions

Field Name	Data Type	Size	Description
EIN or RANGE	String	20	Field can accept two 14 character strings. E.g. 00001234-00003456 for a range
NUMBER OF LEVELS TO EXPLODE	Number	2	Number of levels to display for a particular parent structure.
EXPLOSION QUANTITY	Number	2	Quantity of each EIN to reflect in the report
DATE OF BILL	Date	2	"As of" date used in selecting records from configuration history of the item
TYPE OF SORT	String	1	Null = part, S=Sort string N = Sort number
NOTE 1, NOTE 2	String	40	A 40 character note to include in the report.
EIN MULTI-LEVEL REPORT	Number	2	Number of copies of this report to generate.

Table 27.9.2-2. Procedures to Generate EIN Structure Reports

PERFORM	ACTION
Navigate to EIN Structure Reports screen	From the Main Menu A. Select ILM Main Menu – press 'enter' B. Select ILM Report Menu – press 'enter' C. Select EIN Structure Reports – press 'enter'
Invoke the add command	Press 'a' to go into add mode.
Enter parameters for the report	Fill in the necessary information A. Enter EIN number or zoom to the EIN data file to pick the EIN to report. To do this, press ('/z' , 't' , 'F3'). B. Enter Number of levels to display C. Enter explosion quantity of each EIN to reflect in the report D. Enter the date of Bill used in selecting records E. Enter the type of sort. -null = part -'s' = sort string -'n' =sort # F. Enter any note to appeal on the report header G. Enter number of copies of the report to generate
Exit the add mode	Press 'F3' to exit adds mode.
Execute the report	Press 'e' to execute the report.
Report output	View the report output options: A. Enter 1 and press 'enter' – to view the report on the screen OR B. Enter 2 and press 'enter' – to save the report into a file. Enter the file name to be saved as. The file will be saved in the user's home directory. OR C. Enter 3 and press 'enter' – to print the report to the defaulted printer
Exit the report menu	If option 1 was selected, go to the next step. If option 2 or 3 was selected, press 'F3' to exit the report menu when finished.
Navigate through the report output	A. Navigating through the report -Press 'n' for Next -Press 'p' for Previous -Press 'r' for Right -Press 'q' for Quit or -Press 'h' to print a hardcopy of the report B. After finished making the selection, press 'q' to exit the report screen. A message will come up specifying the number of reports generated, press 'enter' . C. Another message will prompt "Another?" -Press 'y' to generate more reports. This will go back to the EIN Structure Reports screen, or -Press 'n' to go back to the ILM Report Menu.

Remember to press <ENTER> after each field.

15:20 (einstrep) DATE: 01/05/00 TIME

ECS Development Facility EIN STRUCTURE REPORT PAGE:

1 EINs: 00001029 Number of levels:

99 Explosion quantity: 1 Date of bill:

//**

-

Parent EIN: 00001029 Desc: SPARCStation 20-50 SX

MFG Part: S20SX-50-32-P46 Desc: SPARCStation 20-50 SX

Active date: **/**/** Inactive date: **/**/**

LEVEL	EIN	MFG PART	CONTROL ITEM ID	MODEL/VERSION	QUANTITY PER	ACTIVE DATE	INACTIVE DATE
=====	=====	=====	=====	=====	=====	=====	=====
1	00000751	EXB-210TW		210	0.0000	04/12/99	**/**/**
		Tape Stacker - 8 MM					
.2	C0003845	315570-001			0.0000	04/12/99	**/**/**
		BAR CODE READER/EXB-210 & 218					
.2	C0003846	872013-025			0.0000	04/12/99	**/**/**
		8MM Tape Drive					
.2	C0003847	EXB-303220			0.0000	04/12/99	**/**/**
		Terminator					
.2	C0003848	EXB-30726			0.0000	04/12/99	**/**/**
		Tape Cartridge - 8 MM					
.2	C0003849	EXB-307627			0.0000	04/12/99	**/**/**
		Cable - SCSI					
.2	C0003850	TDKP6-1200Q			0.0000	04/12/99	**/**/**
		Tapes - 5 GB - 8 MM					
.2	C0162102	872013-025			0.0000	09/01/99	**/**/**
		8 MM Tape Drive - w/ Carriage Instal					
1	00001086	365-1324-01			0.0000	04/12/99	**/**/**
		20 Inch Color Monitor					
1	00003089	CDE-100		4X	0.0000	04/12/99	**/**/**
		Yamaha External 4X Write/4X Read CD-Rom					
1	00004692	X5511A			0.0000	04/12/99	**/**/**
		2.1 GB HD MultiPack (1 of 2 X 2.1=4.2GB)					
.2	C0021164	540-2730-03			0.0000	04/12/99	**/**/**
		2.1 GB HD Internal					
1	C0147699	SOL		2.4	0.0000	04/22/99	**/**/**
		Solaris					

There are 38 components in this bill.

Figure 27.9.2-2. EIN Structure Report

27.9.3 Install/Receipt Report

This screen (see Figure 27.9.3-1) is designed to allow the user to print a report of a parent EIN configuration and send the hard copy to the receiving organization for sign off. Refer to Figure 27.9.3-2 for a sample report. Table 27.9.3-1 describes the screen's fields, and Table 27.9.3-2 gives the procedure on how to generate the report.

Figure 27.9.3-1. Install/Receipt Report Screen

Table 27.9.3-1. Install/Receipt Report Field Descriptions

Field Name	Data Type	Size	Description
PARENT EIN	String	20	EIN for the parent item in an EIN structure. The operator may zoom to the EIN table and choose the EIN, if it had been entered there previously. (See the EIN Entry section.)
ECS Name through Old User	Multi Fields	30	These fields reflect according to the Parent EIN entered.
INSTALL/RECEIPT REPORT	Number	2	Number of copies of this report to generate.

Table 27.9.3-2. Procedures to Generate Install/Receipt Reports

PERFORM	ACTION
Navigate to Install/Receipt Reports screen	From the Main Menu A. Select ILM Main Menu – press 'enter' B. Select ILM Report Menu - press 'enter' C. Select Install/Receipt Reports - press 'enter'
Invoke the add command	Press 'a' to go into add mode.
Enter report's parameters	Fill in the necessary information A. Enter Parent EIN or zoom to the EIN data file to select the EIN of interest. To do this, press ('/z', 't', 'F3') . B. ECS Name through Old User – these fields are reflected from the Parent EIN you entered above. C. Enter number of copies for the Install/Receipt Report.
Exit the add mode	Press 'F3' to exit add mode.
Execute the report	Press 'e' to execute the report.
Report output	View the report output options: A. Enter 1 and press 'enter' – to view the report on the screen OR B. Enter 2 and press 'enter' – to save the report into a file. Enter the file name to be saved as. The file will be saved in the user's home directory. OR C. Enter 3 and press 'enter' – to print the report to the defaulted printer
Exit the report menu	If option 1 was selected, go to the next step. If option 2 or 3 was selected, press 'F3' to exit the report menu when finished.
Navigate through the report output	A. Navigating through the report -Press 'n' for Next -Press 'p' for Previous -Press 'r' for Right -Press 'q' for Quit or -Press 'h' to print a hard copy of the report B. After finished making the selection, press 'q' to exit the report screen. A message will come up specifying the number of reports generated, press 'enter' . C. Another message will prompt "Another?" -Press 'y' to generate more reports. This will go back to the Install/Receipt Report screen, or -Press 'n' to go back to the ILM Report Menu.

Remember to press <ENTER> after each field.

RUN DATE: 01/05/00

Page No: 1

EOSDIS
EQUIPMENT INSTALLATION/RECEIPT REPORT
BY ECN NUMBER

ECN NUMBER: 00002534
DATE ON-SITE WARRANTY EXPIRES: 12/31/98
WARRANTY END DATE: 12/31/98
HTSC HELP CENTER PHONE: 1-800-ECS-DATA
HTSC HELP CENTER HOURS ARE: 08:00 - 17:00 EST
DATE RECEIVED: 05/09/97

USER CONTACT
USER PHONE
LOCATION: Goddard
BUILDING # GSFC
ROOM #: C101
HOST NAME: g0acs03

I certify that I have received the equipment only for work associated with NASA Contract NAS5 - 60000.

Signature: _____ Date: ____/____/____

MFR	PRODUCT DESCRIPTION	MODEL/VERSION	PART NUMBER	SERIAL NUMBER	PART ECN	INSTALL DATE
SUN	Enterprise 3000 Enc, 4 Slot, CD 4, PWR/C		E3001	715V006C	00002534	05/27/97
WYE	Terminal		900983-07	0ICD6800046	00003256	03/13/98
WYE	Keyboard		901867-01	97030769	00006417	05/27/97
SUN	2.1 GB Internal HD		X5153A	9644628234	C0009199	08/12/99
SUN	250mhz Ultrasparc Modual		2530A	92F30203138	C0014226	05/27/97
SUN	250mhz Ultrasparc Modual		2530A	92F30202448	C0014227	05/27/97
SUN	CPU/Memory Board		2600A	5012976058254	C0014228	05/27/97
SUN	SBUS I/O Board - Enterprise Family		2610A	5014287011120	C0014229	05/27/97
SUN	CD ROM - Internal		370-2203-01	9715003781	C0014230	05/27/97
SUN	9.1 GB HD - 7200 RPM-3.5 Inch-F/W SCSI-2		540-2951-01	9707363003	C0014231	05/27/97
SUN	9.1 GB HD - 7200 RPM-3.5 Inch-F/W SCSI-2		540-2951-01	9707363312	C0014232	05/27/97
SUN	9.1 GB HD - 7200 RPM-3.5 Inch-F/W SCSI-2		540-2951-01	9715742422	C0014233	05/27/97
SUN	32 MB RAM Expansion (1 of 8X32MB=256 MB)		7022A	501265378299929	C0014234	05/27/97
SUN	32 MB RAM Expansion (1 of 8X32MB=256 MB)		7022A	501265378299923	C0014235	05/27/97
SUN	32 MB RAM Expansion (1 of 8X32MB=256 MB)		7022A	501265378299891	C0014236	05/27/97
			.			
			.			
			.			
SUN	Solaris Media for Servers	2.5.1	SOLS-C		C0150689	05/27/97

Figure 27.9.3-2. Equipment Installation/Receipt Report by ECN Number

27.9.4 Installation Summary Reports

This screen (see Figure 29.9.4-1) is designed to retrieve and print all receipts that have occurred for the designated PO, Vendor, or Date. Refer to Figure 27.9.4-2 for a sample report. Table 27.9.4-1 lists the field descriptions. Table 27.9.4-2 provides a set of procedures to generate the report.

installInstallation Summary Reports

File Edit View Options Reports Help

Last: 1 Current: 1

INSTALLATION DATE or RANGE: 01/01/96-12/31/98

NOTE 1: Test 1

NOTE 2: Note 2

ENTER NUMBER OF COPIES

INSTALLATION REPORTS: 1

Text Prior Find Go Select Sort Note Write Execute Help More Quit

Figure 27.9.4-1. Installation Summary Reports Screen

Table 27.9.4-1. Installation Summary Reports Field Descriptions

Field Name	Data Type	Size	Description
INSTALLATION DATE or RANGE	Date	2	Date or range of dates on which installation(s) occurred.
NOTE 1, NOTE 2	String	40	A 40 character note to include in the report.
INSTALLATION REPORTS	Number	4	Number of copies of this report to generate.

Table 27.9.4-1. Procedures to Generate Installation Summary Reports

PERFORM	ACTION
Navigate to Installation Summary Reports Screen	From the Main Menu A. Select ILM Main Menu – press 'enter' B. Select ILM Report Menu - press 'enter' C. Select Installation Summary Reports - press 'enter'
Invoke the add command	Press '/a' to go into add mode.
Enter report's parameters	A. Enter the Installation date or range (i.e. 10/10/98-11/10/98). B. Enter any note to appear on the heading of the report C. Specify number of copies.
Exit the add mode	Press 'F3' to exit the add mode.
Execute the report	Press 'e' to execute the report.
Report output	View the report output options: A. Enter 1 and press 'enter' – to view the report on the screen OR B. Enter 2 and press 'enter' – to save the report into a file. Enter the file name to be saved as. The file will be saved in the user's home directory. OR C. Enter 3 and press 'enter' – to print the report to the defaulted printer
Exit the report menu	If option 1 was selected, go to the next step. If option 2 or 3 was selected, press 'F3' to exit the report menu when finished.
Navigate through the report output	A. Navigating through the report -Press 'n' for Next -Press 'p' for Previous -Press 'r' for Right -Press 'q' for Quit or -Press 'h' to print a hardcopy of the report B. After finished making the selection, press 'q' to exit the report screen. A message will come up specifying the number of reports generated, press 'enter' . C. Another message will prompt "Another?" -Press 'y' to generate more reports. This will go back to the Installation Summary Report screen, or -Press 'n' to go back to the ILM Report Menu.

Remember to press <ENTER> after each field.

(installr)		INSTALLATION SUMMARY REPORT		DATE: 01/05/00	TIME: 13:42	
ECS Development Facility		Dates: 09/01/99-12/31/99		PAGE: 1		
PARENT EIN: 00000343	NAME: judge					
OEM PART: S20SX-50	OEM DESC: SPARCStation 20-50 SX					
INSTALL DATE: 03/10/95						
SITE: EDF ; ECS Development Facility						
BUILDING: 1616 ROOM: 1100A4						
CHILDREN INCLUDED:						
EIN	OEM PART	INSTALL DATE	LOC	BLDG	ROOM	USER
=====	=====	=====	=====	=====	=====	=====
00000343	S20SX-50	03/10/95	EDF	1616	1100A4	Geistfeld

Figure 27.9.4-2. Installation Summary Report

27.10 Maintenance Menu

The ILM Maintenance Menu (see Figure 27.10-1 and Table 27.10-1) is used to maintain the ILM database of maintenance oriented data, generate and track Work Orders for maintenance actions, and schedule preventative maintenance for appropriate items. (Please note) The arrangement of this screen may appear different in ILM but the content within the selections will remain the same.

```
XRP-II - ILM User
ECS Management System
Maintenance Menu
11/20/2000 17:04

maintm

1. Work Order Entry
2. Work Order Modification
3. Preventative Maintenance Items
4. Generate PM Orders
5. Work Order Parts Replacement History
6. Maintenance Work Order Reports
7. Work Order Status Reports
8. Maintenance Codes
9. Maintenance Contracts
10. Authorized Employees
11. Work Order Line Item Query

Please enter selection (1 - 11 or name): 

F1-help F3-prior menu F5-select F8-exit
```

Figure 27.10-1. Maintenance Menu Screen

Table 27.10-1. Maintenance Menu Options

Menu item	Function	Section
Work Order Entry	For entering work orders for repairs	27.10.2
Work Order Modification	For updating work orders as maintenance activity proceeds	27.10.3
Preventative Maintenance Items	For designating which items in the EIN file require preventative maintenance	27.10.4
Generate PM Orders	For generating work orders for items needing preventative maintenance	27.10.5
Work Order Parts Replacement History	For reporting items replaced under one or more work orders.	27.10.6
Maintenance Work Order Reports	For reporting about maintenance activity on selected machines	27.10.7
Work Order Status Reports	For reporting the status of work orders	27.10.8
Maintenance Codes	For defining failure codes to be used when describing repairs and replacements	27.10.9
Maintenance Contracts	For managing information about maintenance contracts with vendors and suppliers	27.10.10
Authorized Employees	For identifying employees permitted access to vendors for repair notification	27.10.11
Work Order Line Item Query	For browsing line item records across multiple maintenance work orders.	27.10.12

27.10.1 Filling out a Maintenance Work Order (MWO)

The purpose of the MWO is to track the following: corrective maintenance, preventative maintenance, configuration change, installation maintenance, and spare replacements. The MWO is the term used to define the formal documentation of maintenance events in a structured manner. The Work Order Entry screen and the Work Order Modification screen are used to initiate and complete the MWO, respectively.

Table 27.10.1-1 specifies the different types of MWOs and how to create them.

NOTE: The LMC only needs to enter a code if the maintenance action is something other than corrective maintenance.

Table 27.10.1-1. Types of MWO

TYPE	CODE	DEFINITION
Corrective Maintenance	CM	Corrective Maintenance (CM) is the unscheduled repair of equipment that includes detection, diagnosis, isolation and resolution through line replaceable unit repair or replacement.
Preventative Maintenance	PM	Preventative Maintenance (PM) is planned maintenance to include routine inspections and servicing that keeps the equipment in good repair in order to prevent failure. To specify the Work Order as a Preventative Maintenance, near the bottom of the MWO screen there is a label call " CODE " enter the code as PM for Preventative Maintenance.
Configuration Change	CC	Configuration change (CC) is the change of the equipment's configuration as directed by an approved CCR. To specify the Work Order as a Configuration Change, near the bottom of the MWO screen there is a label call " CODE " enter the code as CC for Configuration Change Maintenance.
Installation Maintenance	IM	Installation Maintenance (IM) is used for any maintenance action that occurs during the initial installation of new equipment. To specify the Work Order as a Installation Maintenance, near the bottom of the MWO screen there is a label call " CODE " enter the code as IM for Installation Maintenance.
Spare Replacement	SR	Spare replacement result in the use of pre-stocked spared parts. Only in the cases of spare use will two MWOs be created. One work order is used to account for the amount of time required to resolve the problem. A second work order is opened to account for the status of the failed component that the spare replaced. On the second work order specify that the type of MWO is a spare replacement by enter "PR" in the " CODE " field. In addition, on the " NOTE " field, reference the related work order number and who replaced the spare.

27.10.1.2 ECS Hardware States

- Operational – Operational hours are not tracked or reported but are determined by functional string using a formula.
- Impaired – Impaired operations exist when the system performs in less than a fully operational state due to a hardware malfunction. Impaired time is reportable but not chargeable as down time.
- Down – Down time occurs when a system is unable to perform its primary operational function due to a hardware malfunction. Down time is reportable and chargeable. Chargeable down time results only from inherent failures (i.e., no induced failures like power outages) of ECS hardware. Down time starts when the hardware problem is first recognized, and continues until the failed component is 1) repaired, 2) replaced, or 3) switched over to another system.

27.10.2 Work Order Entry Screen

This screen (see Figure 27.10.2-1) is used to initiate the Work Order for repair. User may also initiate a work order through the Work Order Modification screen. This screen is always presented in ADD mode. Refer to Table 27.10.2-1 for the screen field descriptions. Table 27.10.2-2 provides procedure to determine the upper level assembly of failed component and Table 27.10.2-3 gives the procedure to enter a new work order.

Note: XRP-II version 3.1.3 does not copy the bill of materials into the work order item page.

wordent Work Order Entry

[wordent] WORK ORDER ENTRY:

WORK ORDER: RETURN for next

PARENT EIN:

Serial Number:

ECS NAME: Suffix:

OEM Part:

OEM Desc:

Mod/Ver: Location:

Building: Room:

TROUBLE TICKET:

NOTIFICATION DATE: NOTIFICATION TIME:

PRIORITY: SUBMITTER:

FAILURE DATE: FAILURE TIME:

MFR: VENDOR:

MAINT VENDOR:

VENDOR CALL DATE: TIME:

VENDOR CONTACT NAME:

VENDOR REFERENCE:

TEXT:

CODE: NOTE:

CODE: NOTE:

ADD: F1-help F2-clear F3-exit F4-mode F6-default

Typeover mode

Figure 27.10.2-1. Work Order Entry Screen

Table 27.10.2-1. Work Order Entry Field Descriptions (1 of 2)

Field Name	Data Type	Size	Description
WORK ORDER	String	10	This is the actual Work Order number. The operator may press RETURN to obtain the next number sequentially assigned by the system.
PARENT EIN	String	20	EIN for the parent item in an EIN structure. This parent EIN is the next higher assembly EIN number. Refer to table 27.10.2-2 to determine the next higher assembly of the failed component
Serial Number	String	30	Serial Number of the item entered as parent EIN.
Name	String	30	Name of the machine with which the item is associated.
Suffix	String	3	Code which when used as a suffix to ECS Name forms an identifier (RMA ID) for equipment subject to RMA reporting.
OEM Part	String	34	Manufacturer's part number for the item entered as Parent EIN.
OEM Desc	String	30	Manufacturer's description for the item entered as Parent EIN.
Mod/Ver	String	24	Model or version of the item entered as Parent EIN.
Location	String	8	Designator for the inventory location of the item entered as Parent EIN.
Building	String	6	Building where the item entered as Parent EIN is situated.
Room	String	6	Room where the item entered as Parent EIN is situated.
TROUBLE TICKET #	String	15	Identifier for the trouble ticket associated with the work order
NOTIFICATION DATE	Date	2	Date the LMC is notified of the failure. The system defaulted to the current date.

Table 27.10.2-1. Work Order Entry Field Descriptions (2 of 2)

Field Name	Data Type	Size	Description
NOTIFICATION TIME	Time	2	Time the LMC is notified of the failure. The system defaulted to the current time.
PRIORITY	String	1	Priority assigned to the work. 1 being the highest and 3 is the lowest.
SUBMITTER	String	10	Code of the employee who submitted the problem and caused the work order to be opened.
FAILURE DATE	Date	2	The actual failure date. The earliest between the following: 1) the time the LMC is notified or 2) the time the hardware problem is first recognized.
FAILURE TIME	String	2	The actual failure time. The earliest between the following: 1) the time the LMC is notified or 2) the time the hardware problem is first recognized.
MFG/DEV	String	6	Code for the manufacturer. The operator may zoom to the Vendor table and choose the code, if it had been entered previously.
VENDOR	String	6	Code for the vendor from whom the item was purchased.
MAINT VENDOR	String	6	Code for the item's maintenance vendor.
VENDOR CALL DATE	Date	2	Date the vendor was called and informed of the problem.
VENDOR CALL TIME	Time	2	Time the vendor was called and informed of the problem.
VENDOR CONTACT NAME	String	30	Vendor point of contact
VENDOR REFERENCE	String	20	Identifier to be reference when contacting the vendor about the problem with the item
CODE	String	2	Identifier for a type of category of the maintenance action. CM – Corrective Maintenance, PM – Preventative Maintenance, IM – Installation Maintenance, SR – Spare Replacement, and CC – Configuration Change. NOTE: The LMC only needs to enter a code if the maintenance action is something other than corrective maintenance.
NOTE	String	60	A 60 character note that can be associated with this item.
TEXT	String	8	Press /Z at this prompt to obtain a free form text window. The operator may enter the failure / repair details in this window. When complete, press F3 to exit the text window.

Table 27.10.2-2. Determining Upper level assembly of Failed component

PERFORM	ACTION
Navigate to the EIN Manager screen.	From the Main Menu A. Select ILM Main Men – press 'Enter' B. Select EIN Menu – press 'Enter' C. Select EIN Manager – press 'Enter'
Find the component EIN	A. Press ' f ' to invoke the find command B. Enter the 8 digit EIN number of the failed component EIN C. Press ' F5 ' to start the search
Determine the upper level assembly of the failed component	After the system found and displayed the failed component EIN record information on the screen, press ' W ' to view the next upper level assembly. If ILM replied, "No records available" this mean that this item is not attach to any parent structure.
Exit the EIN Manager screen	Press ' F3 ' twice to get out of the EIN Manager screen.

Table 27.10.2-3. Procedures to enter a new work order (1 of 2)

PERFORM	ACTION
Navigate to Work Order Entry screen.	From the Main Menu A. Select ILM Main Menu – press ' enter ' B. Select Maintenance Menu – press ' enter ' C. Select Work Order Entry – press ' enter '

Table 27.10.2-3. Procedures to enter a new work order (2 of 2)

PERFORM	ACTION
Filling out the work order	<p>Fill in the necessary information</p> <p>A. Press 'enter' to get the next work order number.</p> <p>B. Enter Parent EIN number (the next higher assembly of the failed component, refer to table 27.10.2-2 to determine the next higher assembly) or press '/z', choose the Parent EIN from the list by pressing 't', then press 'F3' or "Q" quit</p> <p>C. Serial Number through Room number – these fields reflected from the Parent EIN entered above.</p> <p>D. Enter the applicable trouble ticket number. Entry required only if the HW problem was transferred from a Remedy trouble ticket or NCR.</p> <p>E. Enter the date and time the LMC was notified of the failure.</p> <p>F. Enter problem priority Enter '1' for any malfunction that results in down time of a production system and immediate corrective action is needed. Enter '2' for any malfunction that impairs system performance but does not result in down time; however, may result in down time if system must be brought down to fix the problem. Enter '3' for any malfunction that will not result in system down time (e.g. minor flickering screen, key sticking, sticking mouse, etc...)</p> <p>G. Enter the actual failure date and time. The earliest between the following: 1) the time the LMC is notified or 2) the time the hardware problem is first recognized.</p> <p>H. Enter date and time vendor was called and informed of the problem</p> <p>I. Enter vendor contact name</p> <p>J. Enter maintenance vendor's trouble ticket number. For advance replacement support, enter both the vendor's trouble ticket number and the RMA number, e.g. TT# 123456 RMA # 456789</p> <p>K. Enter the appropriate maintenance code: NOTE: Only need to enter a code if the maintenance action is something other than corrective maintenance. See table 27.10.1-1 for a more detail definition of the maintenance categories. CM – Corrective Maintenance PM – Preventive Maintenance IM – Installation Maintenance SR – Spare Replacement CC – Configuration Change</p> <p>L. Enter any other note about the work order</p>
Exit the add mode	Press 'F3' to exit add mode
Exit the Work Order Entry Screen	Press 'F3' again to exit the Work order Entry screen.

Remember to press <ENTER> after each field.

27.10.3 Work Order Modification Screen

This screen provides the ability to create or modify work orders as maintenance activity proceeds and as additional information about the repair is known. This screen selects work orders with parents located at the local site and precludes users from creating or modifying work orders for equipment that is not located at the local site. It functions much the same as Work Order Entry screen except it can accept more information and can be used to view all work orders pertinent to the site. This screen has a right page (chargeable hours) page on which cumulative downtime data can be recorded. Figure 27.10.3-2 depicts this page and Table 27.10.3-2 describes the fields.

The header screen's Items command provides access to the item page for adding data about components involved in individual maintenance actions. In general, a line item would be created for each component that has failed, been replaced, or been added new. Figure 27.10.3-3 displays the screen and Table 27.10.3-3 describes the fields.

Refer to Figure 27.10.3-1 for the screen display, Table 27.10.3-1 for the field description, Table 27.10.3-5 for a listing of effects on property records by MWO line item processing, and Table 27.10.3-6 for procedure to complete a work order.

Note: XRP-II version 3.1.3 does not copy the bill of materials into the work order item page. The item page should only contain components involved in the maintenance actions.

The screenshot displays the 'wordmodWork Order Modification' screen. The title bar shows 'wordmodWork Order Modification' and the status bar shows 'Last: 1124 Current: 36'. The screen is divided into several sections:

- Work Order Information:** Work Order: EDC0000011, Parent Ein: 00001954, Serial Number: 649F0AD1, Ecs Name: e0dis01, OEM Part: E4000, OEM Desc: Enterprise 4000 Enclosure 8-Slot Card Ca, Mod/Ver: 4000, Building: EROS D, Control Item ID: EDC000000006400.
- Status and Location:** STATUS: A [O=Open A=Audit F=Finish R=Retire], Location: EDC, Room: 1500.
- Dates and Times:** START DATE: 11/14/00, END DATE: 11/14/00, START TIME: 11:00, END TIME: 11:00, REASON: --Maint Tech Init Response--.
- ALDT-1 Section:** START DATE: **/**/, END DATE: **/**/, START TIME: 00:00, END TIME: 00:00, REASON: --ALDT-1--.
- ALDT-2 Section:** START DATE: **/**/, END DATE: **/**/, START TIME: 00:00, END TIME: 00:00, REASON: --ALDT-2--.
- Other Section:** START DATE: **/**/, END DATE: **/**/, START TIME: 00:00, END TIME: 00:00, REASON: --Other (restore OS, etc.)--.
- Events:** Events: T, NOTE: UNRECOVERABLE WRITE ERROR, Date Entered: 11/14/00.
- Navigation:** Next Prior View Find Go Select /Sort /Note Items Help More Quit.

Figure 27.10.3-1. Work Order Modification Screen

Press RIGHT for Down Times -><-<- Press LEFT to return to main page

Table 27.10.3-1. Work Order Modification Field Descriptions (1 of 2)

Field Name	Data Type	Size	Description
WORK ORDER	String	10	Identifier for the work order.
STATUS	String	1	Used for determining status of the MWO. The definition for each of the status is stated below. WARNING: In order to ensure the integrity of the information, it is important that the LMC not update MWOs after the status has been changed to 'A' O – When the LMC opens the MWO. A – When the MWO action is completed the LMC changes the status to 'A' for the ILS Maintenance Coordinator review. F – After ILS Maintenance Coordinator reviewed, the status is changed to 'F'. R – After the ILS Property Administrator reviewed, status is changed to 'R'.
PARENT EIN	String	20	EIN for the parent item in an EIN structure. This parent EIN is the next higher assembly EIN number. Refer to table 27.10.2-1 to determine the next higher assembly of the failed component.
Serial Number through Control Item ID	MULTI-FIELD		These fields are all reflected from the EIN file for the Parent as entered.
TROUBLE TICKET #	String	15	Identifier for the trouble ticket associated with the work order
NOTIFICATION DATE and TIME	MULTI-FIELD		Date and time the problem was reported. These fields are initialized with the current date and time but can be modified.
ALDT REASON CODE	String	10	Code to describe the delays within the maintenance process. AV – Awaiting vendor AP – Awaiting Part UD – User Determined
PRIORITY	String	1	Priority assigned to the work order. 1 being the highest and 3 is the lowest.
FAILURE DATE and TIME	MULTI-FIELD		Actual failure date and time. The earliest between the following: 1) the time the LMC is notified or 2) the time the hardware problem is first recognized.
VENDOR CALL DATE and TIME	MULTI-FIELD		Date and time the maintenance vendor was called.
VENDOR ARRIVE DATE and TIME	MULTI-FIELD		Date and time the vendor arrived to perform the repairs.
VENDOR COMPLETE DATE and TIME	MULTI-FIELD		Date and time the repair was completed whether by on-site support, advanced replacement, spare replacement, or local site personnel repairs.
MFR/DEV	String	6	Code identifying the Manufacturer or Developer ID for the specified parent EIN.

Table 27.10-3-1. Work Order Modification Field Descriptions (2 of 2)

Field Name	Data Type	Size	Description
MAINT VENDOR	String	6	Code identifying the maintenance vendor for the specified parent EIN
VENDOR CONTACT NAME	String	30	Vendor point of contact
VENDOR REFERENCE	String	20	Maintenance vendor's trouble ticket number. For advance replacement support, enter both the vendor's trouble ticket number and the RMA number.
CODE	String	2	Identifier for a type of category of the maintenance action. CM – Corrective Maintenance, PM – Preventative Maintenance, IM – Installation Maintenance, SR – Spare Replacement, and CC – Configuration Change. NOTE: The LMC only needs to enter a code if the maintenance action is something other than corrective maintenance.
EVENTS	String	N/A	Used to enter information relevant to the maintenance event.
NOTE	String	60	This field is used to enter a 60 character note attached to this item.
START DATE and END DATE	MULTI-FIELD		The first block – indicate the time and date that the vendor returns call. Second and third blocks – indicate delay times for when the vendor's work was suspended and resumed (include travel time, admin delays, and logistic delays). Fourth block – indicate the time between when the vendor leaves and the system is back up.
REASON	String	2	Enter the appropriate reason code for the delay entered. AV – Awaiting vendor AP – Awaiting Part UD – User Determined

27.10.3.1 Chargeable Hours Page for Work Order Modification Screens

This screen (see Figure 27.10.3-2 and Table 27.10.3-2) provides the ability to maintain chargeable hours to be used in calculations for downtime.

Field Name	Data Type	Size	Description
ALDT	Floating	10.1	Duration, in hours, of any administrative logistic delays.
TIME TO REPAIR	Floating	10.1	Elapsed time required for technician to complete repair and to bring item to operational status.
SWITCH OVER TIME	Floating	10.1	The time required by the system to restore its functions by switching from the downed equipment to the backup equipment. Switch over time starts at the time of the failure and ends when the fail over system is functional. Enter the total hours (in tenths of an hour).
TOTAL CHARGEABLE DOWNTIME	Floating	10.1	Enter the total hours to be charged for downtime. Specified in tenths of an hour.

Figure 27.10.3-2. Chargeable Hours Page for Work Order Modification Screen

**Table 27.10.3-2. Chargeable Hours Page for Work Order Modification
Field Descriptions**

Field Name	Data Type	Size	Description
ALDT	Floating	10.1	Duration, in hours, of any administrative logistic delays.
TIME TO REPAIR	Floating	10.1	Elapsed time required for technician to complete repair and to bring item to operational status.
SWITCH OVER TIME	Floating	10.1	The time required by the system to restore its functions by switching from the downed equipment to the backup equipment. Switch over time starts at the time of the failure and ends when the fail over system is functional. Enter the total hours (in tenths of an hour).
TOTAL CHARGEABLE DOWNTIME	Floating	10.1	Enter the total hours to be charged for downtime. Specified in tenths of an hour.

27.10.3.2 Items Page for Work Order Modification Screens

This screen provides the ability to add data about components involved in individual maintenance actions. In general, a line item would be created for each component that has failed, been replaced, or been added new. Figure 27.10.3-3 depicts the Left page of this screen and Figure 27.10.3-4 displays the Right page. Refer to Tables 27.10.3-3 and 27.10.3-4 for field descriptions for these screens.

ILS Maintenance Coordinator will review each of the work orders, and the ILS Property Administrator will process any property actions there after.

Note: XRP-II version 3.1.3 does not copy the bill of materials into the work order item page. The item page should only contain components involved in the maintenance actions.

wordmodWork Order Modification

[wordmod] WORK ORDER MODIFICATION: Last: 1124 Current: 36

Work Order: EDC0000011 Status: A Parent Ein: 00001954 Serial Number: 649F0AD1

Ecs Name: e0dis01 Suffix: OEM Part: E4000

Oem Desc: Enterprise 4000 Enclosure 8-Slot Card Ca Mod/ver: 4000

[wl_item] Last: 2 Current: 1

SEQ: 1

COMPONENT EIN: C0029855

--- Database Value --- --- Observed Value ---

OEM PART: 370-1868-03

OEM DESC: Hard Drive - 9 GB Internal

MFR: SUN

MOD/VER:

SERIAL NUMBER: 98081371879

RECEIVE DATE: 02/10/00 ***

Install Date: 11/14/00

Status: I

GFE NUM:

RIGHT for more details ->

EVENT TYPE: F [F=Fail N=New R=Rplc] MAINT CODE: V [R=Relo S=Stock V=Vend]

CHANGE DATE: 11/14/00

Processed?: N

Date Entered: 11/14/00

Failed & Replaced Items Only

FAILURE CODE:

NEW LOCN:

REPLACEMENT'S EIN:

NEW ECS NAME:

BLDG:

ROOM:

USER:

[for Stock/Vend only]

[for Stock/Vend only]

[for Relo only]

Next Prior View Find Go Select /Sort /Note Copy-bill Help More Quit

Figure 27.10.3-3. Items Page (Left) for Work Order Modification Screen (1 of 2)

**Table 27.10.3-3. Items Page (Left) for Work Order Modification (EDF)
Field Descriptions (1 of 3)**

Field Name	Data Type	Size	Description
SEQ	Numeric	4	Number used to distinguish among the line items of a Maintenance Work Order.
COMPONENT EIN	String	20	Identifier for an EIN-controlled item that is a child (component) of a parent EIN and the target of the maintenance event. The operator may zoom to the EIN table to choose an identifier, if it had been entered there previously (see the EIN Manager section). If the field is left null or blank, the system will create an inventory number with a C-prefix for it automatically when the line item is processed.
OEM Part	String	34	Manufacturer's or vendor's part number for the item.
OEM Desc	String	40	Manufacturer's or vendor's description of the item. The operator may zoom to the OEM Parts table to choose a description, if it had been entered there previously (see the OEM Parts section).
MFR	String	6	Code used for the manufacturer of the item. The operator may zoom to the Vendor table to choose a code, if it had been entered there previously (see the Vendor Master section).
MOD/VER	String	24	Model or Version of the item.
SERIAL NUMBER	String	30	Serial number of the item.
RECEIVE DATE	Date	2	Date the item was received.
Install Date	Date	2	Date the item was installed.

**Table 27.10.3-3. Items Page (Left) for Work Order Modification
Field Descriptions (2 of 3)**

Field Name	Data Type	Size	Description
Status	String	1	Code that designates the status of the item. The following values are set when processing transactions: R = Received; S = Shipped; I = Installed; X = Archived;
GFE NUM	String	8	Gov't Furnished Equipment (GFE) number for the item
EVENT TYPE	String	1	Code identifying a type of maintenance event (N=new item installed; F=failed item replaced; R=serviceable item replaced).
MAINT CODE	String	3	Code designating the item's disposition. Property records are updated differently depending on the value entered.
CHANGE DATE	Date	2	Effective date of the configuration change.
EVENTS			A block of free form text for describing maintenance-related activities.
Processed?	String	1	Flag signifying whether or not the line item has been processed by the Work Order's .P(rocess_Changes) bottom-line command. The command updates the Component EIN's property records.
Date Entered	Date	2	Date the line item was created
FAILURE CODE	String	2	Code designating the cause of failure. This code is only used with failed items (i.e., Event Type="F".) The operator may zoom to the Maintenance Codes table and choose the code, if it had been entered there previously. (See the Maintenance Codes section.)
NEW ECS NAME	String	30	ECS name to be recorded in the item's property record. This code is only applicable to items that have failed or are being replaced.
NEW LOCN	String	6	Code for the new inventory location to which the item is to be assigned. This field is used for items that have failed or are being replaced (i.e., Event Type="F" or Event Type="R") and are being returned to stock or to a maintenance vendor. The operator may zoom to the Inventory Locations table to choose a code, if it had been entered there previously (see the Inventory Locations section).
BLDG	String	6	New building where the item is to be installed. This field is used for items that have failed or are being replaced (i.e., Event Type="F" or Event Type="R") and are being returned to stock or to a maintenance vendor. The operator may zoom to the Inventory Locations table to choose a code, if it had been entered there previously (see the Inventory Locations section).
ROOM	String	6	Room where the item is to be installed. This field is used for items that have failed or are being replaced (i.e., Event Type="F" or Event Type="R") and are being returned to stock or to a maintenance vendor.
USER	String	10	New user to which the item is to be assigned. This field is used for items that have failed or are being replaced (i.e., Event Type="F" or Event Type="R") and are being returned to stock or to a maintenance vendor.

**Table 27.10.3-3. Items Page (Left) for Work Order Modification
Field Descriptions (3 of 3)**

Field Name	Data Type	Size	Description
REPLACEMENT'S EIN			Identifier of the new item being used as a replacement. This field is used only for items that have failed or that are being replaced (i.e., Event Type="F", or Event Type="R").
NEW PARENT EIN	String	30	EIN of the item to which the Component EIN is to be re-assigned. This field is applicable only to components that have failed or are being replaced (Event Type="F" or "R"), and are being relocated (Maint Code="R"). <i>The value must be supplied or the item will not get processed.</i> The operator may zoom to the EIN table to choose an identifier, if it had been entered there previously (see the EIN Manager section).

wordmodd Work Order Modification

[wordmodd] WORK ORDER MODIFICATION: Last: 1124 Current: 36

Work Order: EDC0000011 Status: A Parent Ein: 00001954 Serial Number: 649F0AD1

Ecs Name: e0dis01 Suffix: E4000

Oem Desc: Enterprise 4000 Enclosure 8-Slot Card Ca Mod/ver: 4000

[w1_item] Last: 2 Current: 1

SEQ: 1

--- Database Value ---	--- Observed Value ---
MAINT CONTRACT:	
MANF YEAR: 0	0
PO: CCM0005362	
VENDOR: SUN	
WARR EXP DATE: 04/11/01	MM/MM/MM

CODE: NOTE:

CODE: NOTE:

Next Prior View Find Go Select /Sort /Note Copy-bill Help More Quit

Figure 27.10.3-3. Items Page (Right) for Work Order Modification (2 of 2)

Table 27.10.3-4 describes the fields on the Items Page (Right) for Work Order Modification screen.

**Table 27.10.3-4. Items Page (Right) for Work Order Modification
Field Descriptions**

Field Name	Data Type	Size	Description
SEQ	Numeric	4	Number used to distinguish among the line items of a Maintenance Work Order.
MAINT CONTRACT	String	15	Identifier for the maintenance contract as assigned by Purchasing or provided by the vendor.
MANF YEAR	Numeric	4	Date the item was manufactured.
PO NUMBER	String	10	Identifier for the purchase order against which the item was received.
VENDOR	String	6	Code for the vendor from which the item was purchased.
WARRANTY DATE	Date	2	Date the warranty period ends.
CODE	String	2	Identifier for a type or category of note associated with the item.
NOTE	String	60	A 60 character note that can be associated with this item.

The Process_Changes command provides a convenient, reliable, and efficient means for updating ILM property records based on information contained in MWO line items. New EIN records are created as necessary, as are corresponding OEM part, engineering change, and EIN structure records. Processing adds new items to the ECS inventory, archives those that have failed or been returned to the vendor, and re-assigns any that have been relocated or returned to stock. Additionally, items returned to a vendor are rendered obsolete with respect to their parent EINs and, of those that had failed, costs are transferred to their replacements.

If XRP-II is to update property records based on MWO line item data, line item records must specify values for Event Type and Maint Code. They determine the type of property record changes to be made. (See Table 27.10.3-5) Additionally, operators must supply a value for New Parent EIN if an item is designated for relocation. Other line item fields, such as Component EIN, Change Date, Replacement's EIN, New Locn, and New Bldg, have special significance as well in that they influence which database records actually change.

NOTE: Process changes command is reserved for the ILS PA uses only. However, LMC at each DAACs must provide the data detailed in table 27.10.3-5 in order for the ILS PA to process any changes derived from individual work orders.

**Table 27.10.3-5. Effects on Property Records by MWO Line Item Processing
(1 of 3)**

Event Types	Maint Codes	Required Fields	Definition
(F)ailed	(R)elocate	CHANGE DATE NEW PARENT EIN	<p>When an item has failed and been moved to a different machine, the system will:</p> <p>Render the EIN inactive, as of the replacement date, in EIN product structures that use it on or after that date.</p> <p>Update the EIN record for the failed item, clearing its installation date and setting its status to "F", its audit date to the replacement date, and its ECS name and its location values to those of the new parent EIN specified in the line item record.</p> <p>Add the EIN as a component of the new parent EIN as of the replacement date</p> <p>Records an FAI event in the inventory transaction log.</p>
(F)ailed	(V)endor	CHANGE DATE NEW ECS NAME = ARCHIVE NEW LOCN = EDFARC If External item, REPLACEMENT'S EIN	<p>When an item has failed and returned to the vendor, the system will:</p> <p>Render the EIN inactive, as of the replacement date, in EIN product structures that use it on or after that date.</p> <p>Update the EIN record for the failed item, clearing its installation date and setting its status to "X", its audit date to the replacement date, and its ECS name and location values to new values if specified in the line item record.</p> <p>Records an FAI event in the inventory transaction log.</p>
(N)ew	(V)endor	CHANGE DATE	<p>When an item is new and from the vendor, the system will:</p> <p>Render the component inactive in the product structure of other parent EINs, if any.</p> <p>Update the EIN record for the replacement item. Clearing its designator as a spare or consumable and setting its status to "I", its installation, receive, and audit dates to the replacement date, its ECS name to that of the MWO's parent EIN, and its location values to that of the replaced item.</p> <p>If the component is replacing a failed item and the failed item is being returned to the vendor, copy the item cost from the EIN record for the failed item to the EIN record for the new item, then zero out the cost in the EIN record for the failed item</p> <p>Add the component to the product structure of the MWO's parent EIN effective on the replacement date.</p> <p>Records an MRV event in the inventory transaction log.</p>

**Table 27.10.3-5. Effects on Property Records by MWO Line Item Processing
(2 of 3)**

Event Types	Maint Codes	Required Fields	Definition
(R)eplaced	(R)elocate	CHANGE DATE NEW PARENT EIN	<p>When an item is being relocated to a new machine, the system will:</p> <p>Render the EIN inactive, as of the replacement date, in EIN product structures that use it on or after that date.</p> <p>Update the EIN record for the replaced item, clearing its installation date and setting its status to "R", its audit date to the replacement date, and its ECS name and its location values to those of the new parent EIN specified in the line item record.</p> <p>Add the EIN as a component of the specified, new parent EIN as of the replacement date.</p> <p>Records an REP event in the inventory transaction log.</p>
(F)ailed	(S)tock	CHANGE DATE NEW ECS NAME NEW LOCN BLDG ROOM	<p>When an item has failed and returned to stock, the system will:</p> <p>Render the EIN inactive, as of the replacement date, in EIN product structures that use it on or after that date.</p> <p>Update the EIN record for the failed item, clearing its installation date and setting its status to "F", its audit date to the replacement date, and its ECS name and location values to new values if specified in the line item record.</p> <p>Records an FAI event in the inventory transaction log.</p>
(N)ew	(S)tock	CHANGE DATE	<p>When the replacement item is new and taken from stock, the system will:</p> <p>Render the component inactive in the product structure of any other parent EIN (and adjust the inventory count for the losing locations accordingly).</p> <p>Update the EIN record for the replacement item, settings its status to "I", its installation date and audit date to the replacement date, its ECS name to that of the MWO's parent EIN, its location values to that of the replaced item.</p> <p>Add the component to the product structure of the MWO's parent EIN effective on the replacement date.</p> <p>Records an MTR event in the inventory transaction log.</p>

**Table 27.10.3-5. Effects on Property Records by MWO Line Item Processing
(3 of 3)**

Event Types	Maint Codes	Required Fields	Definition
(R)eplaced	(V)endor	CHANGE DATE NEW ECS NAME = ARCHIVE NEW LOCN = EDFARC REPLACEMENT's EIN	<p>When an item is being replaced and returned to the vendor, the system will:</p> <p>Render the EIN inactive, as of the replacement date, in EIN product structures that use it on or after that date.</p> <p>Update the EIN record for the replaced item, clearing its installation date and setting its status to "X", its audit date to the replacement date, and its ECS name and location values to new values if specified in the line item record.</p> <p>Remind the user to adjust item costs manually for the replacement item.</p> <p>Records an REP event in the inventory transaction log.</p>
(R)eplaced	(S)tock	CHANGE DATE NEW ECS NAME NEW LOCN BLDG ROOM REPLACEMENT's EIN	<p>When an item is being relocated and return to stock, the system will:</p> <p>Render the EIN inactive, as of the replacement date, in EIN product structures that use it on or after that date.</p> <p>Update the EIN record for the replaced item, clearing its installation date and setting its status "R", its audit date to the replacement date, and its ECS name and its location values to new values if specified in the line item record.</p> <p>Add the EIN as a component of the specified, new parent EIN as of the replacement date.</p> <p>Records an REP event in the inventory transaction log.</p>

Table 27.10.3-6. Procedures to Complete the Work Order (1 of 4)

PERFORM	ACTION
Navigate to Work Order Modification Screen	From the Main Menu A. Select ILM Main Menu – press 'enter' B. Select Maintenance Menu – press 'enter' C. Select Work Order Modification – press 'enter'
Find the Work Order of interest	A. Press 'f' to find work order to modify B. Enter the work order number and then press 'F5' , or press 'v' to go into the list mode, find the particular record by placing the cursor on the line of the desired record and then press 'v' again to bring back to the individual record mode.
Invoke the modify command	Press '/m' to go into modify mode.
Enter new information about the work order as the maintenance proceed	Fill in the necessary information A. Enter vendor arrive date and time – when the vendor technician arrived on site to perform repair. B. Enter vendor complete date and time whether by on-site support, advanced replacement, spare replacement, or local site personnel repairs. C. At the Events field - Press '/z' to get access to the text box, enter the following information: old and new, part numbers, EIN, and serial number. Other appropriate notes would be any administrative logistics delay times, problems, excessive delays or problem that should be brought to the attention of the ILS office. D. Enter the appropriate maintenance code: NOTE: Only need to enter a code if the maintenance action is something other than corrective maintenance. See table 27.10.1-1 for a more detail definition of the maintenance categories. CM – Corrective Maintenance PM – Preventive Maintenance IM – Installation Maintenance SR – Spare Replacement CC – Configuration Change E. Enter any other note about the work order F. Enter Start Date and End Date First block – indicate the time and date that the vendor returns call Second and third block – indicate delay times for when the vendor's work was suspended and resumed (include travel time, admin delays, and logistic delays). Enter reason for the delay. AV – Awaiting Vendor, AP – Awaiting Part and UD – User Determined. Fourth block – indicate the time between when the vendor leaves and the system is back up. G. Press 'F3' to go to the next step

Table 27.10.3-6. Procedures to Complete the Work Order (2 of 4)

PERFORM	ACTION
Filling in the chargeable hours page if any.	<p>A. Press 'r' to go to the Chargeable Hours page for Work Order Modification screen.</p> <p>B. Press '/m' to go into modify mode</p> <p>C. Enter the Administrative Logistic Delay Time (ALDT).</p> <p>D. Enter elapsed time required for technician to complete repair and to bring item to operational status.</p> <p>E. Enter Switch over time (in tenths of an hour) required for switchover. Switchover time starts at the time of the failure and ends when the failover system is functional.</p> <p>F. Enter the total hours (in tenths of an hour) to be charged for down time.</p> <p>G. Press 'F3' to go to the next step</p>
Invoke the work order's item page	Press 'I' to invoke the items page.
Enter failed or replaced INTERNAL items	<p>If the screen is not in add mode, press '/a' to enter add mode. Enter the following information in the following fields:</p> <p>A. SEQ – press 'enter' – system will generate the next sequence number.</p> <p>B. COMPONENT EIN – Enter C component EIN if known – the system will populate all relevant information about the component on the DATABASE VALUES column. If the C component EIN is not known and is not in the database, fill out the following information on the OBSERVE VALUES column:</p> <p style="padding-left: 40px;">OEM PART – Enter OEM part number – Operator may zoom to oem part table.</p> <p style="padding-left: 40px;">OEM DESC – Enter part description</p> <p style="padding-left: 40px;">MFG – Enter Manufacture code. Operator may zoom to the vendor table.</p> <p style="padding-left: 40px;">SERIAL NUMBER – Enter the item serial number.</p> <p>C. EVENT TYPE – Enter F (Fail) or R (Replace) – Refer to table 27.10.3-5 to determine the appropriate event types.</p> <p>D. MAINT CODE – Enter R (Relocate) or S (Stock) or V (Vendor) - Refer to table 27.10.3-5 to determine the appropriate maintenance codes.</p> <p>E. CHANGE DATE – the actual date the change occurred.</p> <p>F. EVENTS – zoom to the event text block to enter any note about this item.</p> <p>G. Complete the following information only when the item is being relocated to a new parent or a new location within the project. If the item returned to the vendor, leave this section blank.</p> <p style="padding-left: 40px;">NEW LOCN – Enter new location. Operator may zoom to the inventory location table.</p> <p style="padding-left: 40px;">BLDG – Enter new building.</p> <p style="padding-left: 40px;">ROOM – Enter new room.</p> <p style="padding-left: 40px;">NEW PARENT EIN – Enter new parent EIN, if this item is being relocated to a new parent.</p> <p>H. Press 'F3' to exit Add mode.</p>

Remember to press <ENTER> after each field.

Table 27.10.3-6. Procedures to Complete the Work Order (3 of 4)

PERFORM	ACTION
<p>Enter failed or replaced EXTERNAL item</p>	<p>If the screen is not in add mode, press 'a' to enter add mode. Enter the following information in the following fields:</p> <p>A. SEQ – press 'enter' – system will generate the next sequence number.</p> <p>B. COMPONENT EIN – Enter component EIN – the system will populate all relevant information about the component on the DATABASE VALUES column. If the component EIN is not in the database, fill out the following information on the OBSERVE VALUES column: OEM PART – Enter OEM part number – Operator may zoom to oem part table. OEM DESC – Enter part description MFG – Enter Manufacture code. Operator may zoom to the vendor table. SERIAL NUMBER – Enter the item serial number.</p> <p>C. EVENT TYPE – Enter F (Fail) or R (Replace) – Refer to table 27.10.3-5 to determine the appropriate event types.</p> <p>D. MAINT CODE – Enter R (Relocate) or S (Stock) or V (Vendor) - Refer to table 27.10.3-5 to determine the appropriate maintenance codes.</p> <p>E. CHANGE DATE – the actual date the change occurred.</p> <p>F. EVENTS – zoom to the event text block to enter any note about this item.</p> <p>G. Complete the following information only when the item is being relocated to a new parent or a new location within the project. If the item returned to the vendor, leave this section blank. NEW LOCN – Enter new location. Operator may zoom to the inventory location table. BLDG – Enter new building. ROOM – Enter new room. REPLACEMENT 'S EIN – Enter the new EIN that is replacing this item. NEW PARENT EIN – Enter new parent EIN, if this item is being relocated to a new parent.</p> <p>H. Press 'F3' to exit Add mode.</p>

Table 27.10.3-6. Procedures to Complete the Work Order (4 of 4)

PERFORM	ACTION
Enter NEW or REPLACEMENT items	<p>If the screen is not in add mode, press 'a' to enter add mode. Enter the following information in the following fields:</p> <p>A. SEQ – press 'enter' – system will generate the next sequence number.</p> <p>B. COMPONENT EIN – Enter component EIN – If the EIN is in ILM, the system will populate all relevant information about the component on the DATABASE VALUES column. If the component EIN is not known or is not in the database, fill out the following information on the OBSERVE VALUES column:</p> <p style="padding-left: 40px;">OEM PART – Enter OEM part number – Operator may zoom to oem part table.</p> <p style="padding-left: 40px;">OEM DESC – Enter part description</p> <p style="padding-left: 40px;">MFG – Enter Manufacture code. Operator may zoom to the vendor table.</p> <p style="padding-left: 40px;">SERIAL NUMBER – Enter the item serial number.</p> <p>C. EVENT TYPE – Enter N (New)</p> <p>D. MAINT CODE – Enter S (Stock) or V (Vendor) - Refer to table 27.10.3-5 to determine the appropriate maintenance codes.</p> <p>E. CHANGE DATE – the actual date this item was installed.</p> <p>F. Press 'F3' to exit Add mode.</p>
Exit the item page	When finished entering all failed & replacement information, press ' F3 ' to exit the item page. ILM will notify the operator that "there are n number of unprocessed line items." Press ' enter ' to clear the message and return to the MWO modification screen.
Update the MWO status	<p>At the MWO Modification screen</p> <p>A. Press '/m' to invoke the modify mode to change the status of the work order</p> <p>B. Enter an 'A' for audit.</p> <p>C. Press 'F3' to exit the modify mode.</p>
Exit the Work Order Modification screen	Press ' F3 ' to exit Work Order Modification screen.

Remember to press <ENTER> after each field.

27.10.4 Preventative Maintenance Items Screens

The designator of which items in the EIN file has been determined and its frequency entered by the ECS/ILS office. Updates will be based on preventive maintenance 'MWO' submitted by the LMC. See Figure 27.10.4-1 and Table 27.10.4-1.

```

pmemo Preventative Maintenance Items
Last: 54792 Current: 57

EIN: 00000061
Description: SPARCSys LX Workstation
Name: MICKEY
OEM Part: 4-30-6X-32 P46
OEM Desc: SPARCSys LX Workstation
Location: EDF
Building: 1616
Room: 2155D

SET AS PM ITEM (Y/N):
FREQUENCY: 0
LAST DATE: **/**/
MAINTENANCE DUE ON: **/**/

Text Prior View Find Go Select Sort Note Help More Quit

```

Figure 27.10.4-1. Preventative Maintenance Items Screen

Table 27.10.4-1. Preventative Maintenance Items Field Descriptions

Field Name	Data Type	Size	Description
EIN through ROOM	MULTI-FIELD		These fields are not modifiable by the operator and represent the actual data from the EIN file.
SET AS PM ITEM (Y/N):	String	1	Flag designating the item is to undergo preventative maintenance. Y = Yes; N = No
FREQUENCY	Number	3	Number of days between PM.
LAST DATE	Date	2	Last date a PM has performed for this item.
MAINTENANCE DUE ON	String	8	Date the next maintenance is due.

27.10.5 Generate Preventative Maintenance Orders

This screen (see Figure 27.10.5-1 and Table 27.10.5-1) provides the ability to generate Work Orders for item needing a PM (see Table 27.10.5-2 for procedure). When executed, orders are created for all items needing a PM prior to the cutoff date entered and prints a summary report of orders created.

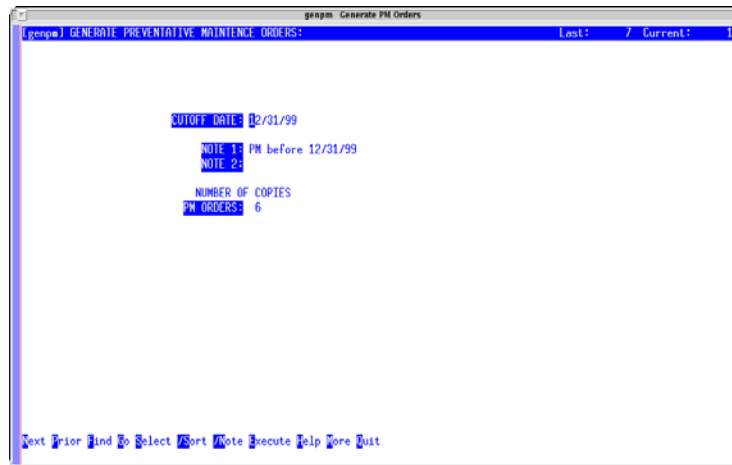


Figure 27.10.5-1. Generate Preventative Maintenance Orders Screen

Table 27.10.5-1. Generate Maintenance Orders Field Descriptions

Field Name	Data Type	Size	Description
CUTOFF DATE	String	8	Enter the last date for the system to examine PM items and generate orders.
NOTE 1 and NOTE 2	String	40	A 40 character note to include in the report
NUMBER OF COPIES (PM ORDERS)	String	1	Number of copies of the report to print

Table 27.10.5-2. Procedures to Generate PM Orders

STEP	ACTION
Navigate to the Generate PM Orders screen	From the Main Menu A. Select ILM Main Menu – press 'enter' B. Select Maintenance Menu – press 'enter' C. Select Generate PM Orders – press 'enter'
Invoke the add command	Press '/a' to go into add mode.
Fill in the parameters to generate the PM orders	Fill in the necessary information A. Enter the last date for the system to examine preventative maintenance items. B. Enter any note to appear on the header of the report C. Enter number of copies of the report
Exit the add mode	Press 'F3' to exit the add mode.
Execute the PM orders	Press 'e' to execute the transaction.
Print the PM orders report	Press 'F3' if you do NOT want to print the report. If you want to print it on the screen, choose option 1 and press 'enter' .
Select report options	A. Make your selection -Press 'n' for Next -Press 'p' for Previous -Press 'r' for Right -Press 'q' for Quit or -Press 'h' to print a hardcopy of the report B. After finished making the selection, press 'q' to exit the report screen. C. A message will come up specifying the number of reports generated, press 'enter' . D. Another message will prompt "Another?" -Press 'y' to generate PM. This will go back to the Generate PM Orders screen, or -Press 'n' , to go back to the Maintenance Menu.

Remember to press <ENTER> after each field.

27.10.6 Work Order Parts Replacement History Screen

The Work Order Parts Replacement History screen (Figure 27.10.6-1) generates reports detailing parts replaced under maintenance work orders. Table 27.10.6-1 describes the screen's fields. The operator enters a Work Order number or range of numbers and a number of copies wanted, then uses the Execute command to print the history reports. Table 27.10.6-2 provides the procedure.

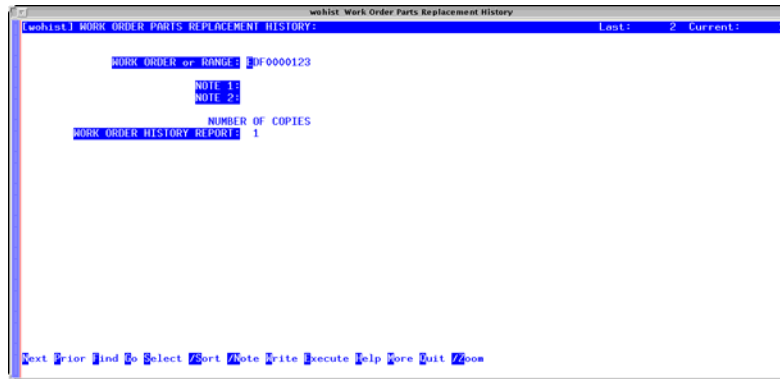


Figure 27.10.6-1. Work Order Parts Replacement History Report

Table 27.10.6-1. Work Order Parts Replacement History Field Descriptions

Field Name	Data Type	Size	Description
WORK ORDER or RANGE	String	25	Identifier for a work order or range of orders.
NOTE 1 and NOTE 2	String	60	A 40-character note to include in the report.
NUMBER OF COPIES (WORK ORDER HISTORY REPORT)	String	1	Number of copies of the report to print

Table 27.10.6-2. Procedures to Generate Work Order Parts Replacement History Report

STEP	ACTION
Navigate to the Work Order Parts Replacement History Report	From the Main Menu A. Select ILM Main Menu – press 'enter' B. Select Maintenance Menu – press 'enter' C. Select Work Order Parts Replacement History – press 'enter'
Invoke the add command	Press '/a' to go into add mode.
Fill in the parameters to generate the report	Fill in the necessary information A. Enter the work order number or range of work orders to report on B. Enter any note to include in the report C. Specify number of copies of the report to generate
Exit the add mode	Press 'F3' to exit the add mode.
Execute the report	Press 'e' to execute the transaction.
Print the report	Press 'F3' if you do NOT want to print the report. If you want to print it on the screen, choose option 1 and press 'enter' .
Select report options	A. Make your selection -Press 'n' for Next -Press 'p' for Previous -Press 'r' for Right -Press 'q' for Quit or -Press 'h' to print a hard copy of the report B. After finished making the selection, press 'q' to exit the report screen. C. A message will come up specifying the number of reports generated, press 'enter' . D. Another message will prompt "Another?" -Press 'y' if you want to generate PM. This will go back to the Generate PM Orders screen, or -Press 'n' , to go back to the Maintenance Menu.

Remember to press <ENTER> after each field.

27.10.7 Maintenance Work Order Reports Screens

This screen (see Figure 27.10.7-1 and Table 27.10.7-1) provides Work Order Reports for work done on selected machines. See Table 27.10.7-2 for procedure.

mwo Maintenance Work Order Reports

Last: 2 Current: 1

PARENT EIN: 00001849

OEM PART:

SERIAL NUMBER:

SITE:

NOTE 1:

ENTER NUMBER OF COPIES

MAINTENANCE WORK ORDER REPORTS: 1

Text Prior Find Go Select Sort Note Execute Help More Quit Zoom

Figure 27.10.7-1. Maintenance Work Order Reports Screen

Table 27.10.7-1. Maintenance Work Order Reports Field Descriptions

Field Name	Data Type	Size	Description
PARENT EIN	String	20	Parent EIN for the installation/structure.
OEM PART	String	34	OEM part number reflected from the EIN record of the child.
SERIAL NUMBER	String	30	Serial number of the parent EIN.
SITE (LOCATION)	String	6	Code for the site which the items can be found.
NOTE 1	String	40	A 40 character message to include in the report
ENTER NUMBER OF COPIES (Maintenance Work Order Reports)	Number	1	Number of copies of the report to print.

Table 27.10.7-2. Procedures to Generate Maintenance Work Order Reports

PERFORM	ACTION
Navigate to the Maintenance Work Order Reports screen	From the Main Menu A. Select ILM Main Menu – press 'enter' B. Select Maintenance Menu – press 'enter' C. Select Maintenance Work Order Reports – press 'enter'
Invoke the add command	Press '/a' to go into add mode.
Enter the report parameters	Fill in the necessary information A. Enter the Parent EIN number, or you may press '/z' , choose the EIN from the list by pressing 't' , then press 'F3' . B. Enter any note to appeal on the header of the report C. Enter number of copies of the report to print.
Exit the add mode	Press 'F3' to exit the add mode.
Execute the report	Press 'e' to execute the transaction.
Print the report	Press 'F3' NOT to print the report. To print it on the screen, choose option 1 and press 'enter' .
Select printing options	A. Make the selection -Press 'n' for Next -Press 'p' for Previous -Press 'r' for Right -Press 'q' for Quit or -Press 'h' to print a hardcopy of the report B. After finished making the selection, press 'q' to exit the report screen. C. A message will come up specifying the number of reports generated, press 'enter' . D. Another message will prompt "Another?" -Press 'y' to print more report. This will go back to the Maintenance Work Order reports screen, or -Press 'n' , to go back to the Maintenance Menu.

Remember to press <ENTER> after each field.

27.10.8 Work Order Status Reports Screens

This screen (see Figure 27.10.8-1 and Table 27.10.8-1) provides status reports on selected Work Orders. See Table 27.10.8-2 for procedure to generate maintenance work order reports.

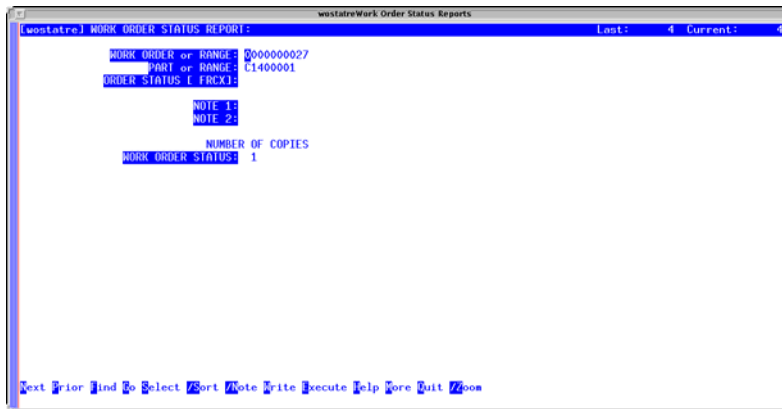


Figure 27.10.8-1. Work Order Status Reports Screen

Table 27.10.8-1. Work Order Status Reports Field Descriptions

Field Name	Data Type	Size	Description
WORK ORDER or RANGE	String	25	Identifier for a work order or a range of orders. The operator may zoom to the Work Order file to choose an identifier, if it had been entered there previously. (See the Work Order Entry section.)
PART (OEM PART NUMBER) or RANGE	String	34	Manufacturer's part number or a range of numbers for items. The operator may zoom to the OEM Part file to choose the part number, if it had been entered there previously. (See the OEM Part Numbers section.)
ORDER STATUS [FRCX] (STATUS)	String	2	Code for the status of a work order
NOTE 1, NOTE 2	String	40	A 40 character message to include in the report
ENTER NUMBER OF COPIES (Work Order Status)	Number	1	Enter any number of copies of the report to print.

Table 27.10.8-2. Procedures to Generate Maintenance Work Order Reports

PERFORM	ACTION
Navigate to the Work Order Status Reports Screen	From the Main Menu A. Select ILM Main Menu – press 'enter' B. Select Maintenance Menu – press 'enter' C. Select Work Order Status Reports – press 'enter'
Invoke the add command	Press '/a' to go into add mode.
Enter the report parameters	Fill in the necessary information. This report can be run without filling in any fields except "NUMBER OF COPIES," Fill in selection criteria for the other fields as required. A. Enter the Work order number, or you may press '/z' , choose the work order from the list by pressing 't' , then press 'F3' . B. Enter the child OEM part number - you may press '/z' , choose the OEM Part from the list by pressing 't' , then press 'F3' . C. Enter order status O - OPEN – when the order is first entered A – AUDIT – when the order is being reviewed by the ILS MC. F – FINISH – when the order is being reviewed by the ILS PA. R – RETIRE – when the order is closed. D. Enter any note to appeal on the header of the report E. Enter number of copies of the report to print. F. Press 'F3' to exit the add mode.
Execute the report	Press 'e' to execute the transaction.
Print the report	Press 'F3' if you do NOT want to print the report. If you want to print it on the screen, choose option 1 and press 'enter' .
Select printing options	A. Make your selection -Press 'n' for Next -Press 'p' for Previous -Press 'r' for Right -Press 'q' for Quit or -Press 'h' to print a hardcopy of the report B. After finished making your selection, press 'a' to exit the report screen. C. A message will come up specifying the number of reports generated, press 'enter' . D. Another message will prompt up "Another?" -Press 'y' if you want to print more report. This will take you back to the Work Order Status reports screen, or -Press 'n' , this will take you back to the Maintenance Menu.

Remember to press <ENTER> after each field.

PLEASE NOTE the Maintenance Menu options for the following are reserved for the ILS Maintenance Coordinator and ILS Property Administrators. The view option is available for information only. The ILS Maintenance Coordinator and Property Administrators will maintain the add/modify/deletion for these options.

1. Maintenance Codes
2. Maintenance Contracts
3. Authorized Employees

27.10.9 Maintenance Codes Screens

This screen (see Figure 27.10.9-1 and Table 27.10.9-1) provides failure codes and descriptions for use with repairs and replacements.

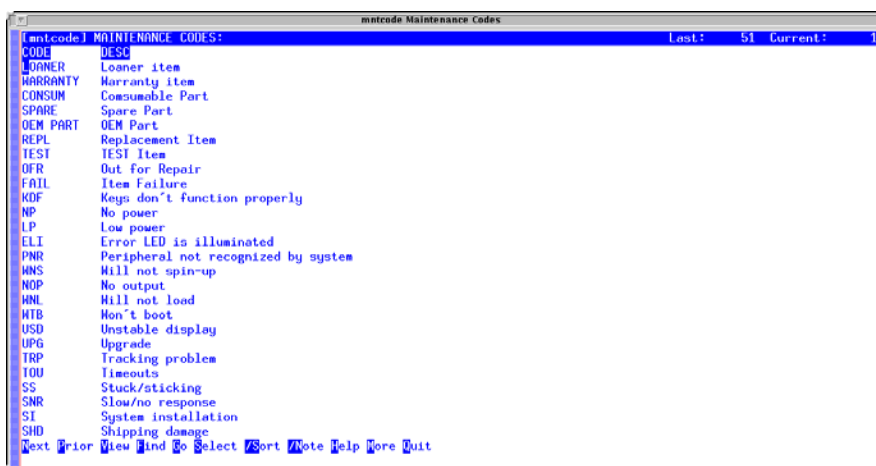


Figure 27.10.9-1. Maintenance Codes Screen

Table 27.10.9-1. Maintenance Codes Field Descriptions

Field Name	Data Type	Size	Description
CODE (maintenance)	String	2	Code that distinguishes among item failures according to their cause
DESC (maintenance)	String	30	Description for the failure code.

27.10.10 Maintenance Contracts Screens

This screen (see Figure 27.10.10-1 and Table 27.10.10-1) provides contract numbers for repair contracts with vendors and suppliers. This screen is maintained by the ILS Maintenance Coordinator.

mntcont Maintenance Contracts

[mntcont] MAINTENANCE CONTRACTS: Last: 62 Current: 1

CONTRACT ID: CCH9718 Zoom to related items

VENDOR: SGI

START DATE: 04/01/98

END DATE: 10/31/02

PO NUMBER: CCH0009718

PIN: 16X7, ON-SITE, 4HR

COMMENT:

Next Prior View Find Go Select /Sort /Note Help More Quit /Zoom

Figure 27.10.1-1. Maintenance Contracts Screen

Table 27.10.10-1. Maintenance Contracts Field Descriptions

Field Name	Data Type	Size	Description
CONTRACT ID	String	15	Identifier for the maintenance contract as assigned by purchasing or provided by the vendor
VENDOR	String	6	Vendor code whom the item was purchased from.
START DATE	Date	2	Date the contract is to become effective.
END DATE	Date	2	Date the contract will expire.
PO NUMBER	String	10	Identifier for the purchase order under which maintenance was procured.
PIN	String	20	PIN number applicable for authorization for vendor contact.
COMMENT	String	60	Comment about the maintenance contract.

27.10.11 Authorized Employees Screens

This screen (see Figure 27.10.11-1 and Table 27.10.11-1) provides employee codes for employees who have been permitted access to the vendor for repair notification.

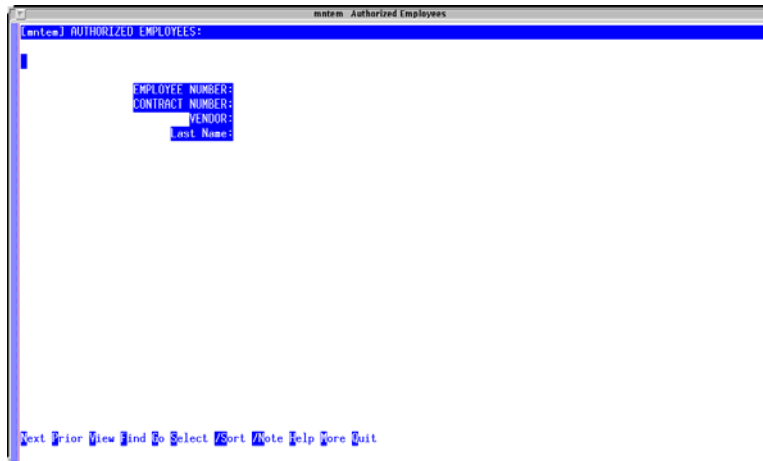


Figure 27.10.11-1. Authorized Employees Screen

Table 27.10.11-1. Authorized Employees Field Descriptions

Field Name	Data Type	Size	Description
EMPL	String	10	Identifier for an employee. The operator may zoom to the Employee table and choose the code, if it had been entered there previously. (See the Employee Manager section.)
CONTRACT NO	String	10	Identifier for maintenance contract. . The operator may zoom to the Maintenance Contracts table and choose the identifier, if it had been entered there previously. (See the Maintenance Contracts Manager section.)
VENDOR	String	6	Code for the vendor with whom the contract is placed. The operator may zoom to the Vendor table and choose the code, if it had been entered there previously. (See the Vendor Master Maintenance section.
LAST NAME	String	30	Last name of the employee. The value is obtained from the Employee table.

27.10.12 Work Order Line Item Query Screen

The Work Order Line Item Query screen provides the ability to browse line items for all Maintenance Work Orders. Operators can use this screen to find and select all work orders under which maintenance actions have been performed for specific component EINs. This screen has left and right pages (Figures 27.10.12-1 and 27.10.12-2) that are nearly identical to the items pages for Work Order Modification. Refer to Tables 27.10.3-3 and 27.10.3-4 above for the descriptions of these fields.

Wlq Work Order Line Item Query Last: 591 Current: 1

Work Order: 000000031
Seq: 1
Component Ein: C0005438
--- Database Value --- --- Observed Value ---
Dem Part: PS-RAID-SP
Dem Desc: RAID Controller
Mfr/dev: SGI
Mod/Ver:
Serial Number: MT002326397 MT002326397
Rec Date: 11/21/96 11/21/96 RIGHT for more details ->
Install Date: 03/28/00
Status: X
Gfe Num:
----- Maintenance Activity -----
Event Type: R [F=Fail N=New R=Rplc] Maint Code: [R=Relo S=Stock V=Vend]
Change Date: 07/08/99 Events: T
Processed?: Y Date Entered: **/**/**
----- Failed & Replaced Items Only -----
Failure Code: New Ecs Name: [for Stock/Vend only]
New Locn: Bldg: Room: [for Stock/Vend only]
Replacement Ein: User:
New Parent Ein: [for Relo only]

Next Prior View Find Go Select /Sort /Note Help More Quit

Figure 27.10.12-1. Work Order Line Item Query (Left page) (1 of 2)

Wlq Work Order Line Item Query Last: 202 Current: 202

Work Order: 000000038
Seq: 9
--- Database Value --- --- Observed Value ---
Maint Contract: 1993 0
Purchase Order: FFT0000021
Vendor: DEC
WARR EXP DATE: 12/31/97 **/**/**

Next Prior View Find Go Select /Sort /Note Help More Quit

Figure 27.10.12-2. Work Order Line Item Query (Right Page) (2 of 2)

27.11 License Menu

Many software products used in ECS are licensed; that is, subject to conditions of limiting how many users can run the product and where. Licenses take numerous forms. Nodelock licenses let users run the product, but only on a designated machine; counted nodelock licenses limit the

number of users that can run the product on that machine. Floating licenses allow users to run a product on any machine in a network. They may limit the number of users that can run the product concurrently, the number of servers that can be used concurrently, the number of sites that can use the product, or any combination of the above. Licenses can apply to a named product, one or more of its features, one or more of its versions, and/or one or more types of platforms. Some vendors enforce these provisions through use of license keys, but ECS is accountable for adhering to licensing provisions whether vendors use keys or not.

The life cycle for licensed COTS software encompasses developmental and systems engineering, purchasing, receiving, stocking, distribution, installation, use, and recovery. Licenses associated with COTS products are obtained, allocated, and archived; they also expire. Allocations can be re-assigned and recovered. Licenses do not always change when the licensed product does.

Multiple licenses are sometimes obtained from the product vendor under the provisions of a single license certificate. Each license would account for part of the rights-to-use under the certificate. Conversely, individual licenses can consume rights-to-use from more than one certificate. Each unique license key implies a unique license, but not every license has a key.

Licenses are allocated to the sites and host machines where their keys are installed, and keyless licenses are allocated to where their software products are installed. This is not so much for property accounting (i.e., cost accounting), but to verify adherence to purchased licensing provisions and to identify where licenses are used in case rights-to-use must be transferred elsewhere. A single license can be allocated to multiple sites and machines.

License rights-to-use is counted differently depending on the type of licenses purchased. Rights for nodelock license are allocated and counted by node and are consumed at the rate of one license per node. Floating license rights are allocated and counted based on number of users on a network rather than by specific machines, where the network is represented by a machine on which the license is installed. Floating license rights are consumed at the rate of number of users per license. Occasionally, a purchased entitlement covers a total number of users across a limited number of machines. In this case, rights are consumed at the rate of one license per node as well as number of users per license.

The License Menu (Figure 27.11-1 and Table 27.11-1) provides access to XRP-II's capability for managing software licenses.

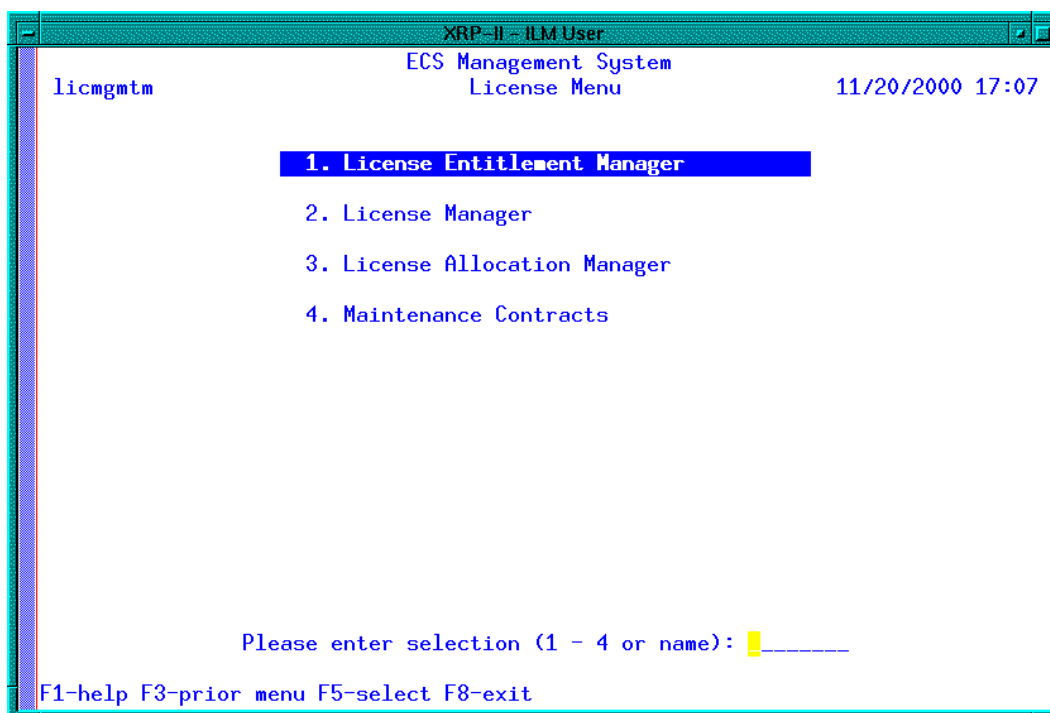


Figure 27.11-1. License Menu

Table 27.11-1. ILM Master Menu Options

Menu item	Function	Section
License Entitlement Manager	For browsing purchased license entitlements at the DAACs.	27.11.1
License Manager	For tracking software license keys licenses obtained from vendors	27.11.2
License Allocation Manager	For tracking license rights-to-use allocated to hosts and sites;	27.11.3
Maintenance Contracts	For managing information about maintenance contracts with vendors and suppliers.	27.10.10

Note: Software License Management is managed by the Software License Administrator at EDF. DAACs only have query and report generating capabilities.

27.11.1 License Entitlement Manager Screen

Operators use the License Entitlement Manager screen (Figure 27.11.1-1 and Table 27.11.1-1) to maintain records of purchased rights-to-use for licensed software, including how many node and user rights-to-use have been consumed, remain, and are under maintenance. An entitlement record usually corresponds to a line item on a purchase order much like an EIN for hardware, but it can also represent rights associated with one or more copies of a software product for which

licenses are not purchased separately. A single record can accommodate a mix of both node and user rights-to-use. Rights consumed and remaining are computed automatically (and on demand) based on the licenses mapped against it.

The screenshot shows a terminal window titled "Licmtd License Entitlement Manager". The top status bar indicates "Last: 936" and "Current: 188". The main content area displays the following information:

```

LICENSE ENTITLEMENT ID: C0143038
ENT OEM PART: CLEARCASE
Ent Description: ClearCase Single user License
Sw Prod Alias: Clearcase

LICENSE TYPE: FLOATING          Hdusft Code: SE

----- Node Rights -----
RIGHTS TO USE (RTU): 0
ALLOCATED: 0
REMAINING: 0
UNDER MAINT: 0

----- User Rights -----
RIGHTS TO USE (RTU): 17
ALLOCATED: 17
REMAINING: 0
UNDER MAINT: 0

VENDOR: ASC                      Vendor Name: Atria Software Corp
PO NUMBER: CCM0004581             DATE RECEIVED: 06/26/96
STATUS CODE: R                   WARRANTY EXP DATE: 10/31/02
MAINT CONTRACT: CCM4581
COMMENT:

CODE:      NOTE:
CODE:      NOTE:
  
```

At the bottom, a command line shows various navigation and action options: `Next Prior View Find Go Select /Sort /Note .Adjust_qtys .Licenses Help More Quit /Zoom`.

Figure 27.11.1-1. License Entitlements Manager Screen

The following bottom-line commands are unique to this screen:

- **.Adjust_qtys** - updates how many of the license entitlement's node and user rights-to-use are currently allocated and how many remain. This function is useful because quantities are adjusted automatically only when license allocation data is changed via the data entry screens.
- **.Licenses** - activates an items page that lists all the licenses associated with the entitlement.

Table 27.11.1-1. License Entitlement Manager Field Descriptions

Field Name	Data Type	Size	Description
LICENSE ENTITLEMENT ID	String	20	Identifier for a purchased license entitlement. The equivalent of an EIN number.
ENT OEM PART	String	34	Manufacturer's or vendor's part number for the entitlement.
Ent Description	String	40	Manufacturer's or vendor's description for the entitlement. This field reflects the description of the OEM Part Number entered in the field above.
Sw Prod Alias	String	40	Common name used in ECS for the licensed product and all its versions and variants.
LICENSE TYPE	String	16	Classification that distinguishes among licenses according to rules of use. Examples include: floating (limited number of concurrent users), nodelocked (limited to use on a single machine), user (limited to use by a certain individual), project (unlimited use anywhere by individuals working on a certain project), site (unlimited use at a single site), etc.
Hdwsft Code	String	10	Code for classifying inventory items by type. For license entitlements, the code defaults to SE (Software Entitlement).
Rights to Use (RTU)	Numeric	8	Quantity of node or user rights-to-use authorized by this purchased entitlement.
Allocated	Numeric	8	Quantity of node or user rights under the license entitlement currently allocated by licenses mapped to the entitlement. This value is calculated by the system and reflects the total number of active allocations of those licenses.
Remaining	Numeric	8	Quantity of node or user rights under a license entitlement not yet consumed by the mapping of licenses to the entitlement.
UNDER MAINT	Numeric	8	Quantity of node or user rights-to-use currently under maintenance.
VENDOR	String	6	Code for the vendor from whom the item was purchased.
Vendor Name	String	30	Name of the vendor from whom the item was purchased.
PO NUMBER	String	10	Identifier of the purchase order against which the item was received.
STATUS CODE	String	1	Code that designates the status of the software product. The following values are set when processing transactions: R = Received; S = Shipped; I = Installed; X = Archived;
DATE RECEIVED	String	8	Date item was received from vendor.
MAINT CONTRACT	String	15	Identifier for the Maintenance Contract under which the item is covered.
WARRANTY EXP DATE	Date	2	Date the warranty on the entitlement ends. This field defaults to 365 days from the date of entry.
CODE	String	2	Identifier for a type or category of note associated with the item
NOTE	String	60	A message that can be associated with the item.

The **.L** command invokes the Entitlement-Licenses items page depicted in Figure 27.11.1-2. This screen and its cousin, the Licenses-Entitlement items page attached to the License Manager screen, serve the same purpose: to map licenses obtained from vendors to the entitlements whose rights-to-use they consume. Multiple licenses may be mapped to a single entitlement, and a single license may be mapped to multiple entitlements. The Node RTU Allocated and User RTU Allocated fields specify how many of each type of rights a license draws from the entitlement and are what is used by the system when calculating an entitlement's rights consumed. The screen ensures that:

- a) the rights-to-use attributed to the entitlement do not exceed the entitlement's rights remaining;
- b) the sum of the rights being attributed for a license across multiple entitlements do not exceed the rights-to-use for the license.

The screenshot shows a window titled "licmnt License Entitlement Manager (EDF)". It contains two main sections. The top section, titled "License Entitlement Manager (EDF)", shows a table with two rows: "License Id: 0000002" and "Entitlement Ein: C0143038". The bottom section, titled "ENTITLEMENT - LICENSES:", shows details for the license. It includes fields for "License Id: 0000002", "Start Date: 05/11/00", "End Date: 10/31/02", "Entitlement Ein: C0143038", "Sw Prod Alias: Clearcase", "Vendor: ASC", and "Maint Contract: CCM4581". Below these are fields for "Lic Key Node Rtu: 0", "Node Rtu Remain: 10", "Lic Key User Rtu: 205", "User Rtu Remain: 3", "NODE RTU ALLOCATED: 0", and "USER RTU ALLOCATED: 17". At the bottom, there is a menu bar with options: "Next", "Prior", "View", "Find", "Go", "Select", "Sort", "Note", "Help", "More", "Quit", and "Zoom".

Figure 27.11.1-2. Entitlement – Licenses Page

Table 27.11.1-2 describes the fields on the Entitlement-Licenses Page.

Table 27.11.1-2. Entitlement – Licenses Page Field Descriptions

Field Name	Data Type	Size	Description
License Id	String	20	Unique designator for a license.
Start Date	Date	2	Date on which the license record takes effect.
End Date	Date	2	Date on which the license record is rendered ineffective
Entitlement Ein	String	20	Identifier for a purchased license entitlement. The equivalent of an EIN number.
Sw Prod Alias	String	40	Common name used in ECS for the licensed product and all its versions and variants.
Vendor	String	6	Code for the Vendor from whom the license entitlement was purchased.
Maint Contract	String	15	Identifier for the Maintenance Contract under which the license entitlement is covered.
Lic Key Node Rtu	Numeric	8	Number of machines on which the licensed product may be run concurrently sharing the same license key, if any. This value limits how many host allocation records can be created for the license.
Lic Key User Rtu	Numeric	8	Number of users authorized by the license to run the licensed product concurrently on a single network. This value limits the user rights-to-use that can be recorded in the license's allocation records.
Node Rtu Remain	Numeric	8	Quantity of node rights under a license entitlement not yet consumed by the mapping of licenses to the entitlement.
User Rtu Remain	Numeric	8	Quantity of user rights under a license entitlement not yet consumed by the mapping of licenses to the entitlement.
NODE RTU ALLOCATED	Numeric	8	Number of node rights-to-use to be counted under the entitlement as having been consumed by the license. The value may not exceed the current value plus the node rights remaining under the entitlement.
USER RTU ALLOCATED	Numeric	8	Number of user rights-to-use to be counted under the entitlement as having been consumed by the license. The value may not exceed the current value plus the user rights remaining under the entitlement.

27.11.2 License Manager Screen

The License Manager screen (Figure 27.11.2-1 and Table 27.11.2-1) maintains records of software licenses obtained from vendors. Licenses can be mapped to purchase license entitlements so that consumption of license rights can be tracked. A license may also be mapped to individual sites and hosts in order to track allocations, but only after it has first been mapped to one or more entitlements. This helps preclude allocating rights that exceed entitlements purchased.

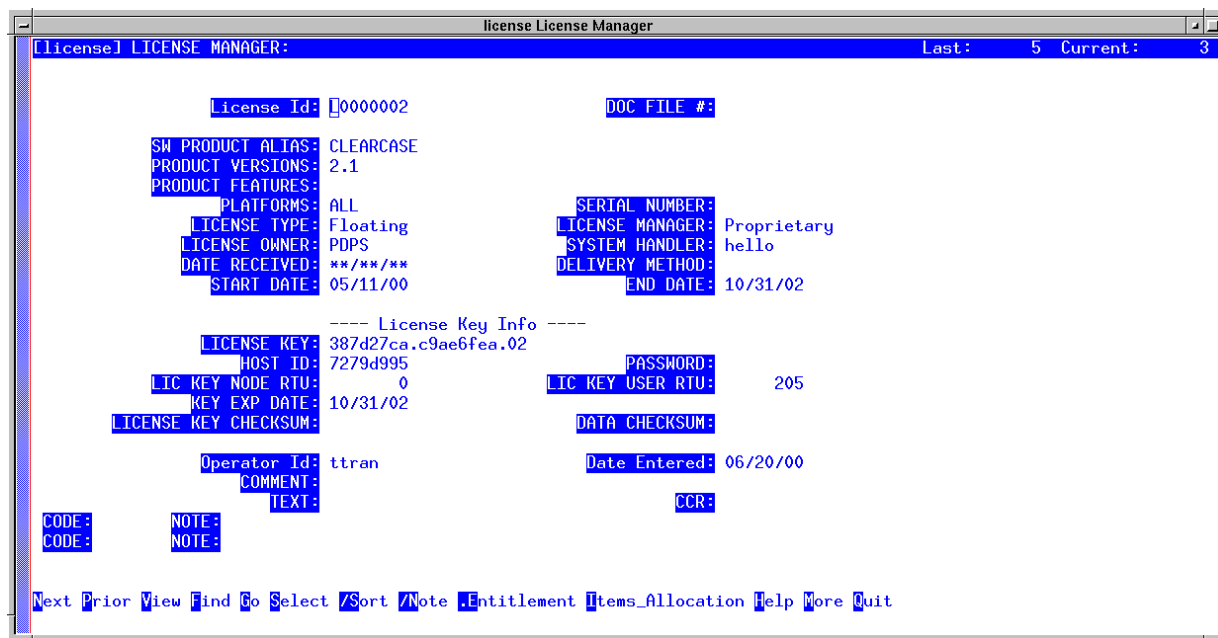


Figure 27.11.2-1. License Manager Screen

The following bottom-line commands are unique to this screen:

- **.Entitlement** - activates an items page that lists the purchased entitlements from which the license's rights-to-use are derived.
- **Items_Allocation** - activates an items page that lists the host machines and sites to which the license has been allocated. The license's rights-to-use must have first been mapped to at least one entitlement before the license can be allocated.

Table 27.11.2-1. License Manager Field Descriptions (1 of 2)

Field Name	Data Type	Size	Description
License Id	String	20	Unique designator for a license.
DOC FILE #	String	20	Identifier under which any hardcopy records or correspondence pertaining to the license have been filed.
SW PRODUCT ALIAS	String	40	Common name used in ECS for the licensed product and all its versions and variants.
PRODUCT VERSIONS	String	24	Identifier(s) of one or more versions of the licensed product that are covered by the license.
PRODUCT FEATURES	String	54	Name(s) of one or more features of the licensed product that are covered by the license.
PLATFORMS	String	15	One or more codes for the types of machines to which the license applies (e.g., Sun, SGI, PC, etc.)
SERIAL NUMBER	String	30	Vendor-supplied serial number for the license or the product being licensed.
LICENSE TYPE	String	16	Classification that distinguishes among licenses according to rules of use. Examples include: floating (limited number of concurrent users), nodelocked (limited to use on a single machine), user (limited to use by a certain individual), project (unlimited use anywhere by individuals working on a certain project), site (unlimited use at a single site), etc.
LICENSE MANAGER	String	12	Technology employed in managing the license on-line (e.g., flexlm, proprietary, etc.)
SYSTEM HANDLER	String	30	Name of the system handler as provided by the license vendor.
DATE RECEIVED	Date	2	Date the license key and/or data arrived.
DELIVERY METHOD	String	10	Means by which the license key and/or data arrived (e.g., mail, e-mail, fax, etc.)
START DATE	Date	2	Date on which the license record takes effect. As of its end date, neither the license nor any of its associated allocations are counted in computations of node or user rights against entitlements. Changing the start date causes earlier start dates in allocation records to be changed to match.
END DATE	Date	2	Date on which the license record is rendered ineffective. This is not the same as the license expiration date. As of its end date, neither the license nor any of its associated allocations are counted in computations of node or user rights against entitlements. Changing the end date causes later end dates in allocation records to be changed to match.

Table 27.11.2-1. License Manager Field Descriptions (2 of 2)

Field Name	Data Type	Size	Description
LICENSE KEY	String	50	String of alphanumeric characters that represent the provisions for a license in an encoded form.
HOST ID	String	20	Host id of the license server machine supplied to the vendor when requesting the license. This is an information only field. Allocations of licenses to machines are accomplished via the License Allocation Manager screen.
PASSWORD	String	20	Password supplied along with the license key by the vendor. This is an information only field.
LIC KEY NODE RTU	Numeric	8	Number of machines on which the licensed product may be run concurrently sharing the same license key, if any. This value limits how many host allocation records can be created for the license.
LIC KEY USER RTU	Numeric	8	Number of users authorized by the license to run the licensed product concurrently on a single network. This value limits the user rights-to-use that can be recorded in the license's allocation records.
KEY EXP DATE	Date	2	Date on which the license key is no longer usable. This is not the same as the license end date, which is the date the license is no longer needed or used. The key expiration date is not used in computing license rights consumed against entitlements.
LICENSE KEY CHECKSUM	String	10	Checksum of the license key as supplied by the license vendor. (Checksums are used by the vendors to verify that a key was copied and matches what was issued.
DATA CHECKSUM	String	10	Checksum for license data supplied by the vendor.
Operator Id	String	8	Login id of the user who created the record.
Date Entered	Date	2	Date the record was created.
COMMENT	String	60	Comment to be stored in the record.
TEXT	String	n/a	A block of text associated with the current record.
CCR	String	10	Identifier for the CCR authorizing the license.
CODE	String	2	Identifier for a type or category of note associated with the item
NOTE	String	60	A message that can be associated with the item.

The License - Entitlements page (Figure 27.11.2-2) manages the mapping of a license to purchased entitlements and specifies how many node and/or user rights-to-use the license is consuming from each. The screen ensures that:

- the rights-to-use attributed to an entitlement do not exceed the entitlement's rights remaining;
- the sum of the rights being attributed to all entitlements do not exceed the rights-to-use for the license.

license License Manager																			
[License] LICENSE MANAGER:		Last:	5 Current: 3																
License Id:	L0000002	Doc File #:																	
Sw Product Alias: CLEARCASE																			
[lie_ent] LICENSE - ENTITLEMENTS:		Last:	3 Current: 1																
<table border="0"> <tr> <td>Entitlement Ein:</td> <td>C0143038</td> <td></td> <td></td> </tr> <tr> <td>Sw Prod Alias:</td> <td>Clearcase</td> <td></td> <td></td> </tr> <tr> <td>Vendor:</td> <td>ASC</td> <td></td> <td></td> </tr> <tr> <td>Maint Contract:</td> <td>CCW4581</td> <td></td> <td></td> </tr> </table>				Entitlement Ein:	C0143038			Sw Prod Alias:	Clearcase			Vendor:	ASC			Maint Contract:	CCW4581		
Entitlement Ein:	C0143038																		
Sw Prod Alias:	Clearcase																		
Vendor:	ASC																		
Maint Contract:	CCW4581																		
Lic Key Node Rtu:	0	Lic Key User Rtu:	205																
Node Rtu Remain:	10	User Rtu Remain:	3																
NODE RTU ALLOCATED:	0	USER RTU ALLOCATED:	17																

Next Prior View Find Go Select /Sort /Note Help More Quit /Zoom

Figure 27.11.2-2. License – Entitlements Page

Table 27.11.2-2 describes the fields on the License-Entitlements Page.

Table 27.11.2-2. License – Entitlements Page Field Descriptions

Field Name	Data Type	Size	Description
Entitlement Ein	String	20	Identifier for a purchased license entitlement. The equivalent of an EIN number.
Sw Prod Alias	String	40	Common name used in ECS for the licensed product and all its versions and variants.
Vendor	String	6	Code for the Vendor from whom the license entitlement was purchased.
Maint Contract	String	15	Identifier for the Maintenance Contract under which the license entitlement is covered.
Lic Key Node Rtu	Numeric	8	Number of machines on which the licensed product may be run concurrently sharing the same license key, if any. This value limits how many host allocation records can be created for the license.
Llic Key User Rtu	Numeric	8	Number of users authorized by the license to run the licensed product concurrently on a single network. This value limits the user rights-to-use that can be recorded in the license's allocation records.
Node Rtu Remain	Numeric	8	Quantity of node rights under a license entitlement not yet consumed by the mapping of licenses to the entitlement.
User Rtu Remain	Numeric	8	Quantity of user rights under a license entitlement not yet consumed by the mapping of licenses to the entitlement.
NODE RTU ALLOCATED	Numeric	8	Number of node rights-to-use to be counted under the entitlement as having been consumed by the license. The value may not exceed the current value plus the node rights remaining under the entitlement.
USER RTU ALLOCATED	Numeric	8	Number of user rights-to-use to be counted under the entitlement as having been consumed by the license. The value may not exceed the current value plus the user rights remaining under the entitlement.

The License Allocations items page (see Figure 27.11.2-3) maintains records about the hosts and sites to which software licenses have been allocated, and it has its own items page, License Allocation Additional Hosts, for identifying redundant or backup server machines on which this license will be installed.

One license allocation record is required for each host on which the license is installed where rights are to be counted as consumed. Allocations to hosts that are redundant or backup server machines are not typically counted against license entitlements and can be recorded as additional hosts associated with the allocation to the primary server.

The screen helps prevent licenses from being over-allocated. A license may not be allocated until it has first been mapped to at least one license entitlement, and allocations may not exceed the rights-to-use reflected in the license record or in the mappings to associated entitlements. In other words, a license cannot be allocated to:

- 1) more hosts than specified by:
 - a) the license's Node Rights-To-Use;

- b) the sum of all Node RTU Allocated in corresponding License – Entitlement records;
- 2) more users than specified by:
 - a) the license’s User Rights-To-Use;
 - b) the sum of all User RTU Allocated in corresponding License – Entitlement records.

The screen also helps operators determine if their license and license allocation records are consistent with the current name, location, and status in the EIN record for that host. An inconsistency suggests that a license-related issue may exist that should be resolved.

The following bottom-line commands are unique to this screen:

- **Items_Addl** – This command activates an items page that lists the backup or redundant server hosts for the allocated license. These items are not included when calculating the rights-to-use allocated and remaining for purchased entitlements.

The screenshot shows a terminal window titled "license license Manager". The main content area displays the following information:

```

License Manager:
License Id: 10000002 Doc File #:
Sw Product Alias: CLEARCASE
Licall111 LICENSE ALLOCATIONS: Last: 1 Current: 1

Alloc Sequence: 1
Sw Product Alias: CLEARCASE
License Key: 387d2/ca.c9ae6fea.02

ALLOC HOST NAME: KRYPTON ALLOC HOST ID: 7279d995
ALLOC SITE: EDF ALLOC STATUS: I
USER RIGHTS TO USE: 205
START DATE: 05/11/00 END DATE: 10/31/02

---Host EIN Data ---
HOST EIN: 00001438
Hostname: KRYPTON
Location: EDF
Status Code: I

Operator Id: xrpadm Date Entered: 08/30/00
COMMENT:
TEXT:

CODE: NOTE:
CODE: NOTE:
  
```

At the bottom of the window, a command line contains the following text: `Next Prior View Find Go Select /Sort /Note Items_Addl Help More Quit`.

Figure 27.11.2-3. License Allocations Page

Table 27.11.2-3 describes the fields on the License Allocations Page screen.

Table 27.11.2-3. License Allocations Page Field Descriptions

Field Name	Data Type	Size	Description
Alloc Sequence	Numeric	4	Number used for identifying uniquely the principal host allocation records for a specific license.
SW PRODUCT ALIAS	String	40	Common name used in ECS for the licensed product and all its versions and variants.
License Key	String	50	String of alphanumeric characters that represent the provisions for a license in an encoded form.
ALLOC HOST NAME	String	30	ECS name of a machine to which the license is allocated.
ALLOC HOST ID	String	8	Host id of a machine to which a license is allocated.
ALLOC SITE	String	6	Code for the site to which the license is allocated.
ALLOC STATUS	String	1	Implementation status of the license with respect to the host or site.
CODE	String	2	Identifier for a type or category of note associated with the item
NOTE	String	60	A message that can be associated with the item.
USER RIGHTS TO USE	Numeric	8	Quantity of user rights being consumed for this license allocation.
START DATE	Date	2	Date on which the license allocation takes effect. Computations of node and user rights consumed against entitlements do not include any associated with allocations having a start date after the current date
END DATE	Date	2	Date on which the allocation of the license to the host expires. This is not the same as the license expiration date. As of its end date, an allocation is no longer counted in computations of user or node rights against entitlements.
HOST EIN	String	30	EIN number of the host to which the license is allocated.
Hostname	String	30	Name of the machine with which the Host EIN is associated.
Location	String	8	Identifier that designates the inventory location of the Host EIN.
Status Code	String	1	Code that designates the status of the Host EIN. The following values are set when processing transactions: R = Received; S = Shipped; I = Installed; X = Archived.
Operator Id	String	8	Login id of the user who created the record.
Date Entered	Date	2	Date the record was created.
COMMENT	String	60	Comment to be stored in the record.
TEXT	String	n/a	A block of text associated with the current record.

The License Allocation Additional Hosts screen (Figure 27.11.2-4) maintains records about backup or redundant license servers for machines to which a license has been allocated. Identifying additional hosts has no effect on calculations of entitlements' node or user rights-to-use consumed or remaining, but is useful for tracking where licenses are supposed to be or may be installed. As a convenience, the screen lets operators specify a Host EIN to facilitate corroborating license allocation data with data in ILM property records.

The screenshot shows a window titled "license License Manager". It contains two main sections. The top section displays license information: "License Id: L0000002", "Doc File #:", "Alloc Sequence: 1", "Sw Product Alias: CLEARCASE", "License Key: 387d27ca.c9ae6fea.02", and "Last: 5 Current: 3". The bottom section, titled "LICENSE ALLOCATION ADDITIONAL HOSTS", shows details for "Sequence: 1": "ALLOC HOST ID: 807e79e6", "ALLOC HOST NAME: PELICAN", "HOST EIN: 00000057", "Hostname: PELICAN", "Status: R", and "Location: EDF". At the bottom of the window is a menu bar with options: "Next", "Prior", "View", "Find", "Go", "Select", "Sort", "Note", "Help", "More", and "Quit".

Figure 27.11.2-4. License Allocation Additional Hosts

Table 27.11.2-4 describes the fields on the License Allocation Additional Hosts screen.

Table 27.11.2-4. License Allocation Additional Hosts Field Descriptions

Field Name	Data Type	Size	Description
Sequence	Numeric	4	Number used for identifying uniquely the records that describe the backup or redundant license servers for a machine allocated a specific license.
ALLOC HOST ID	String	20	Host id of a machine that is a backup or redundant license server for the one to which the license is principally allocated.
ALLOC HOST NAME	String	30	ECS name of a machine that is a backup or redundant license server for the one to which the license is principally allocated.
HOST EIN	String	30	EIN number of the host to which the license is allocated.
Hostname	String	30	Name of the machine with which the Host EIN is associated.
Status	String	1	Code that designates the status of the Host EIN. The following values are set when processing transactions: R = Received; S = Shipped; I = Installed; X = Archived.
Location	String	8	Identifier that designates the inventory location of the Host EIN.

27.11.3 License Allocation Manager Screen

The License Allocation Manager screen (Figure 27.11.3-1) maintains records about the hosts and sites to which software licenses have been allocated. The screen is a near clone of the License Allocations items page of the License Manager screen, permitting operators to browse and update all allocation records at once rather than one license at a time. Refer to Section 27.11.2-3 for the description.

licall License Allocation Manager

Selected: 80 Current: 14

License Id: 0000002
Alloc Sequence: 1
Sw Product Alias: CLEARCASE
License Key: 387d27ca.c9ae6fea.02

ALLOC HOST NAME: KRYPTON
ALLOC SITE: EDF
USER RIGHTS TO USE: 205
START DATE: 05/11/00
END DATE: 10/31/02

ALLOC HOST ID: 7279d995
ALLOC STATUS: I

---- Host EIN DATA ----
HOST EIN: 00001438
Hostname: KRYPTON
Location: EDF
Status Code: I

Operator Id: xrpadm
Date Entered: 08/30/00

COMMENT:
TEXT:

CODE:
NOTE:

Next Prior View Find Go Select /Sort /Note Items_Addl Help More Quit /Zoom

Figure 27.11.3-3. License Allocation Manager

27.11.4 Pre-defined License reports

ILM has three pre-defined software license reports that provide useful information to the License Administrator and the DAACs. Refer to Sections 27.11.4.1 through 27.11.4.3 for more detail information about these reports.

27.11.4.1. License Entitlements Status Report

License Entitlements Status report provides the license purchased, such as: number of rights to use purchased, number of rights remaining, special flags that indicate whether the number allocated was over consumed and whether maintenance warranty expired or will expire within 30 days. The report is sorted by product name. This report is attached to the License Entitlement Manager report's menu. Refer to Table 27.11.4.1-1 for field descriptions, and Table 27.11.4.1-2 for procedure to generate the report.

See Figure 27.11.4.1-1 for a sample display of the License Entitlements Status report.

Table 27.11.4.1-1. License Entitlement Status Report Field Descriptions

Field Name	Data Type	Size	Description
License Type	String	16	Classification that distinguishes among licenses according to rules of use. Examples include: floating (limited number of concurrent users), node locked (limited to use on a single machine), user (limited to use by a certain individual), project (unlimited use anywhere by individuals working on a certain project), site (unlimited use at a single site), etc.
EIN	String	20	Identifier for a purchased license entitlement. The equivalent of an EIN number.
OEM DESC	String	40	Manufacturer's or vendor's description for the entitlement. This field reflects the description of the OEM Part Number entered in the field above.
VENDOR	String	6	Code for the vendor from whom the item was purchased.
PO NUMBER	String	10	Identifier of the purchase order against which the item was received.
MAINT CONTRACT	String	15	Identifier for the Maintenance Contract under which the item is covered.
WARRANTY EXP DATE	Date	2	Date the warranty on the entitlement ends. This field defaults to 365 days from the date of entry.
User RTU	Numeric	8	Quantity of user rights-to-use authorized by this purchased entitlement.
URTU REM	Numeric	8	Quantity of user rights under a license's entitlement not yet consumed by the mapping of licenses to the entitlement.
URTU MNT	Numeric	8	Quantity of user rights-to-use currently under maintenance.
NODE RTU	Numeric	8	Quantity of node rights-to-use authorized by this purchased entitlement.
NRTU REM	Numeric	8	Quantity of node rights under a license's entitlement not yet consumed by the mapping of licenses to the entitlement.
NRTU MNT	Numeric	8	Quantity of node rights-to-use currently under maintenance. An asterisk (*) is an indicated of the flag.
U	Flag		Flag designates user rights to use was over allocated. An asterisk(*) is an indicated of the flag.
N	Flag		Flag designates node rights to use was over allocated.
M	Flag		Flag designates maintenance warranty expired or will expire within 30 days. An asterisk (*) is an indicated of the flag.

Table 27.11.4.1-2. Procedures to Generate License Entitlement Status Reports

PERFORM	ACTION
Navigate to the License Entitlement Manager Screen	From the Main Menu A. Select ILM Main Menu – press 'enter' B. Select License Menu – press 'enter' C. Select License Entitlement Manager – press 'enter'
Selecting data to report.	You may select a subset of the license entitlements to report on, or you may run the report for all the license entitlements available.
To activate the report menu	A. Press '/r' to display the report menu. Note: The number of selections on the report option may be difference according to each of the screen, However, the License Entitlement screen has the following report options: 1. TABLE REPORT 2. FORM REPORT 3. ASCII REPORT 4. LICENSE ENTITLEMENTS STATUS REPORT B. Select option 4 (License Entitlements status report) and press 'enter' . A report destination option box will appear.
Displaying the report	Make your selection on where to display the report. The report destination has three options: 1. CRT – to display the report on the screen. 2. FILE – save the report to a file. You can find the file in your home directory. 3. Ltr-land-80 – print the report directly to your default printer.
Return to the License Entitlement Manager screen	Press 'F3' until XRP takes you back to the License Entitlement Manager Screen.

Remember to press <ENTER> after each field.

(liestatr)
ECS Development Facility
Sw Prod Alias: Clearcase*

LICENSE ENTITLEMENTS STATUS

DATE: 12/01/00 TIME: 13:17
PAGE: 1

*** Clearcase ***

LIC TYPE	EIN	OEM DESC	VENDOR	PURCHASE ORDER	MAINT CONTRACT	WARR DT	USER RTU	URTU REM	URTU MNT	NODE RTU	NRTU REM	NRTU MNT	U	N	M
	C0147636	ClearCase 60 user License	ASC	SG-254515-	N/A	12/31/98	0	0	0	0	0	0			*
	C0156262	ClearCase 60 user License	ASC	SG-254519-	N/A	12/31/02	0	0	0	0	0	0			
FLOATING	C0146938	ClearCase Single user License	ASC	CCM0001631	CCM1631	10/31/02	180	0	0	0	0	0			
FLOATING	C0143038	ClearCase Single user License	ASC	CCW0004581	CCW4581	10/31/02	17	0	0	0	0	0			
FLOATING	C0147677	ClearCase Single user License	ASC	CCW0004528	CCW4528	06/30/98	100	19	0	0	0	0			*

Figure 27.11.4.1-1. License Entitlements Status Report

27.11.4.2 License Allocation by Product Report

License Allocation by Product Report identifies each of the products and their associated license information if any. As the name indicates, this report is sorted by product. The report lists the License Number, seq, host name, host id, license key, expiration date, user rtu, start date, and end date. It also lists redundant hosts if any. This report is attached to the License Allocation Manager screen report menu (Section 27.11.3). Please refer to Table 27.11.4.2-1 for the field descriptions, and Table 27.11.4.2-2 for procedure to generate the report.

See Figure 27.11.4.2-1 for a sample report.

Table 27.11.4.2-1. License Allocations by Product Field Descriptions

Field Name	Data Type	Size	Description
LICENSE	String	20	Unique designator for a license.
SEQ	Numeric	4	Number used for identifying uniquely the principal host allocation records for a specific license.
ALLOC HOST NAME	String	30	ECS name of a machine to which the license is allocated.
ALLOC HOST ID	String	8	Host id of a machine to which a license is allocated.
LICENSE KEY	String	50	String of alphanumeric characters that represent the provisions for a license in an encoded form.
KEY EXP DATE	Date	2	Date on which the license key is no longer usable.
PLATFORMS	String	15	One or more codes for the types of machines to which the license applies (e.g., Sun, SGI, PC, etc.)
USER RTU	Numeric	8	Quantity of user rights being consumed for this license allocation.
START DATE	Date	2	Date on which the license allocation takes effect.
END DATE	Date	2	Date on which the allocation of the license to the host expires.

Table 27.11.4.2-2. Procedures to Generate License Allocations by Product Reports

PERFORM	ACTION
Navigate to the License Entitlement Manager Screen	From the Main Menu A. Select ILM Main Menu – press ' enter ' B. Select License Menu – press ' enter ' C. Select License Allocation Manager – press ' enter '
Selecting data to report.	Operator may choose whether the report is to list all or only active allocations, certain software products, and certain sites' data.
To activate the report menu	A. Press ' /r ' to display the report menu. Note: The number of selections on the report option may be difference according to each of the screen, However, the License Entitlement screen has the following report options: 1. TABLE REPORT 2. FORM REPORT 3. ASCII REPORT 4. LICENSE ALLOCATIONS BY PRODUCT REPORT 5. LICENSE ALLOCATIONS BY HOST REPORT B. Select option 4 and press ' enter '. A report destination option box will appear.
Displaying the report	Make your selection on where to display the report. The report destination has three options: 1. CRT – to display the report on the screen. 2. FILE – save the report to a file. You can find the file in your home directory. 3. Ltr-land-80 – print the report directly to your default printer.
Return to the License Allocation Manager screen	Press ' F3 ' until XRP takes you back to the License Allocation Manager Screen.

Remember to press <ENTER> after each field.

*** Clearcase ***			Versions: 2.1		Features:							
LICENSE	SEQ	HOST NAME	HOST ID	LICENSE KEY	EXP DATE	PLATFORMS	USER RTU	START DT	END DATE			
=====	=====	=====	=====	=====	=====	=====	=====	=====	=====			
L0000038	1	KRYPTON	7279d995	387d27ca.c9ae6fea.02	**/**/**	ALL	0	**/**/**	**/**/**			
L0000039	1	tlms04	807fe113	387d2a2e.01ee9020.02	**/**/**	ALL	0	**/**/**	**/**/**			
-- Addl Host:	2	tlms02	80804996	387d2a2e.01ee9020.02	**/**/**	ALL	0	**/**/**	**/**/**			
-- Addl Host:	1	tlcss01	80857bb0	387d2a2e.01ee9020.02	**/**/**	ALL	0	**/**/**	**/**/**			
L0000040	1	p0mss02	8080bfb7	387e7228.528a5aad.02	**/**/**	ALL	0	**/**/**	**/**/**			
L0000041	1	p0mss02	8080bfb7	387e717b.f4b05dab.02	**/**/**	ALL	0	**/**/**	**/**/**			
L0000042	1	p0mss02	8080bfb7	387e70be.9600a857.02	**/**/**	ALL	0	**/**/**	**/**/**			
L0000043	1	p0mss02	8080bfb7	387e702e.79f1f224.02	**/**/**	ALL	0	**/**/**	**/**/**			
L0000044	1	m0mss02	8080fd39	387d28fe.3da6012e.02	**/**/**	ALL	0	**/**/**	**/**/**			
L0000045	1	g0mss02	8080c9e8	387d287f.95f02f0b.02	**/**/**	ALL	0	**/**/**	**/**/**			
-- Addl Host:	3	g0css02	7279e28d	387d287f.95f02f0b.02	**/**/**	ALL	0	**/**/**	**/**/**			
-- Addl Host:	2	g0mss10	8080f61b	387d287f.95f02f0b.02	**/**/**	ALL	0	**/**/**	**/**/**			
-- Addl Host:	1	g0mss07	807b10bc	387d287f.95f02f0b.02	**/**/**	ALL	0	**/**/**	**/**/**			
L0000046	1	e0mss02	8080f3d1	387e7122.577caf6e.02	**/**/**	ALL	0	**/**/**	**/**/**			
L0000047	1	l0mss01	80806745	387e6f6a.d56ca8fd.02	**/**/**	ALL	0	**/**/**	**/**/**			
-- Addl Host:	2	l0css02	7278fdeb	387e6f6a.d56ca8fd.02	**/**/**	ALL	0	**/**/**	**/**/**			
-- Addl Host:	1	l0mss05	807b1107	387e6f6a.d56ca8fd.02	**/**/**	ALL	0	**/**/**	**/**/**			
L0000048	1	n0mss02	808579b7	387d2a01.03aa9d75.02	**/**/**	ALL	0	**/**/**	**/**/**			
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----			

Figure 27.11.4.2-2. License Allocations by Product Report

27.11.4.3 License Allocations by Host Report

License Allocations by Host Report provides license information for each host or selected hosts depends on operator preferences. The report is sorted by host name and product name. For each host, it lists the host name, host id, host status, machine information, and license information. It also lists any redundant host that the license was assigned. Refer to Table 27.11.4.3-1 for detail descriptions of each of the fields. This report can be found in the License Allocation Manager report menu. Table 27.11.4.3-2 gives the instruction to generate the report, and Figure 27.11.4.3-1 displays a sample report.

Table 27.11.4.3-1. License Allocations by Host Report Field Descriptions

Field Name	Data Type	Size	Description
HOST NAME	String	30	ECS name of a machine to which the license is allocated.
HOST ID	String	8	Host id of a machine to which a license is allocated.
ALLOC STATUS	String	1	Implementation status of the license with respect to the host or site.
EIN Through DESC			These fields populated according to the entered EIN number.
LICENSE	String	20	Unique designator for a license.
SEQ	Numeric	4	Number used for identifying uniquely the principal host allocation records for a specific license.
ECS ALIAS	String	40	Common name used in ECS for the licensed product and all its versions and variants.
VERSIONS	String	24	Identifier(s) of one or more versions of the licensed product that are covered by the license.
PLATFORMS	String	15	One or more codes for the types of machines to which the license applies (e.g., Sun, SGI, PC, etc.)
License Key	String	50	String of alphanumeric characters that represent the provisions for a license in an encoded form.
USER RTU	Numeric	8	Quantity of user rights being consumed for this license allocation.
START DATE	Date	2	Date on which the license allocation takes effect.
END DATE	Date	2	Date on which the allocation of the license to the host expires.

Table 27.11.4.3-2. Procedures to Generate License Allocations by Host Reports

PERFORM	ACTION
Navigate to the License Entitlement Manager Screen	From the Main Menu A. Select ILM Main Menu – press ' enter ' B. Select License Menu – press ' enter ' C. Select License Allocation Manager – press ' enter '
Selecting data to report.	Operator may choose whether the report is to list all or only active allocations, all hosts, and certain hosts data.
To activate the report menu	A. Press ' /r ' to display the report menu. Note: The number of selections on the report option may be difference according to each of the screen, However, the License Entitlement screen has the following report options: 1. TABLE REPORT 2. FORM REPORT 3. ASCII REPORT 4. LICENSE ALLOCATIONS BY PRODUCT REPORT 5. LICENSE ALLOCATIONS BY HOST REPORT B. Select option 5 and press ' enter '. A report destination option box will appear.
Displaying the report	Make your selection on where to display the report. The report destination has three options: 1. CRT – to display the report on the screen. 2. FILE – save the report to a file. You can find the file in your home directory. 3. Ltr-land-80 – print the report directly to your default printer.
Return to the License Allocation Manager screen	Press ' F3 ' until XRP takes you back to the License Allocation Manager Screen.

Remember to press <ENTER> after each field.

*** p0mss02 ***		ALLOC HOSTID: 8080bfb7		ALLOC STATUS: I					
EIN: 00001821		EIN HOSTID:		EIN STATUS: I		MFR: SUN	MODEL: 1170	SERIAL NUMBER: 647F1185	
DESC: Ultra Server 2 w/1-167MHZ CPU,128MB									
LICENSE	SEQ	ECS ALIAS	VERSION	PLATFORMS	LICENSE KEY	EXP DATE	USER RTU	START DT	END DATE
=====	=====	=====	=====	=====	=====	=====	=====	=====	=====
L0000040	1	Clearcase	2.1	ALL	387e7228.528a5aad.02	**/**/**	0	**/**/**	**/**/**
L0000041	1	Clearcase	2.1	ALL	387e717b.f4b05dab.02	**/**/**	0	**/**/**	**/**/**
L0000042	1	Clearcase	2.1	ALL	387e70be.9600a857.02	**/**/**	0	**/**/**	**/**/**
L0000043	1	Clearcase	2.1	ALL	387e702e.79f1f224.02	**/**/**	0	**/**/**	**/**/**
*** tlmss04 ***		ALLOC HOSTID: 807fel13		ALLOC STATUS: I					
EIN: 00003317		EIN HOSTID:		EIN STATUS: I		MFR: SUN	MODEL:	SERIAL NUMBER: 716F0632	
DESC: Ultra 2 Model 2170 w/2 167 MHZ CPU									
LICENSE	SEQ	ECS ALIAS	VERSION	PLATFORMS	LICENSE KEY	EXP DATE	USER RTU	START DT	END DATE
=====	=====	=====	=====	=====	=====	=====	=====	=====	=====
L0000039	1	Clearcase	2.1	ALL	387d2a2e.01ee9020.02	**/**/**	0	**/**/**	**/**/**
-- Addl Host:	1	tlcss01							
-- Addl Host:	2	tlmss02							

Figure 27.11.4.3-1. License Allocations by Host Report

27.12 ILM Master Menu

The ILM Master Menu (see Figure 27.12-1 and Table 27.12-1) provides the LMC with the ability to verify the accuracy of ILM in reference to employee information, inventory location, hardware and software codes and OEM Part Numbers. All additions, deletions or modifications are to be directed to the ILS Property Administrator.

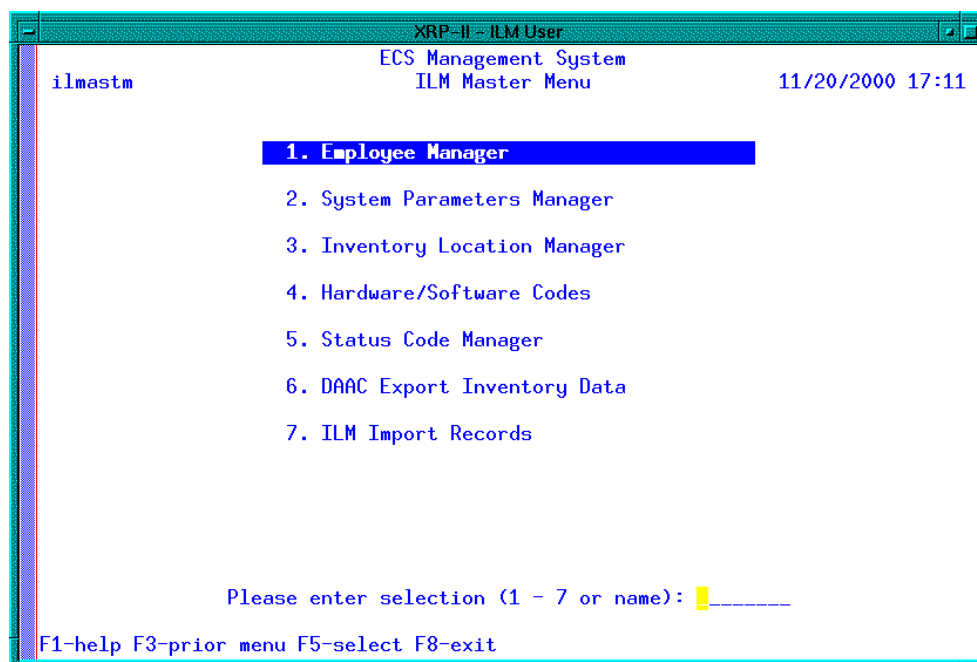


Figure 27.12-1. ILM Master Menu

The ILM Master menu is broken down into the following functions:

Table 27.12-1. ILM Master Menu options

Menu item	Function	Section
Employee Manager	For maintaining employee information	27.12.1
System Parameters Manager	For creating parent/child relationship between components in an assembly	27.12.2
Inventory Location Manager	For maintaining standardized information about ECS inventory locations for all ILM processes.	27.12.3
Hardware/Software Codes	For maintaining a standard set of codes for classifying inventory items according to type.	27.12.4
Status Code Manager	For maintaining a standard set of codes for classifying inventory items according to status.	27.12.5
DAAC Export Inventory Data	For exporting a DAAC's inventory data and transferring them to the SMC's ILM system.	27.12.6
ILM Import Records	For uploading inventory data that had been exported at another site.	27.12.7

27.12.1 Employee Manager Screens

This screen (see Figure 27.12.1-1 and Table 27.12.1-1) provides for the maintenance of employee information.

The screenshot shows a window titled "sfempmtEmployee Manager". The top status bar displays "Last: 883 Current: 11". The main area contains the following fields and values:

- EMPLOYEE NUMBER: 008
- LAST NAME: Fisher
- FIRST NAME: Edson
- STATUS: A
- WORK CENTER: 1
- PHONE: (301)883-4123
- FAX NUMBER:
- E-MAIL:
- PAGER NUMBER:
- CC MAIL:

At the bottom, there is a menu bar with the following options: Text, Prior, View, Find, Go, Select, Sort, Note, Help, More, Quit.

Figure 27.12.1-1. Employee Manager Screen

Table 27.12.1-1. Employee Manager Field Descriptions

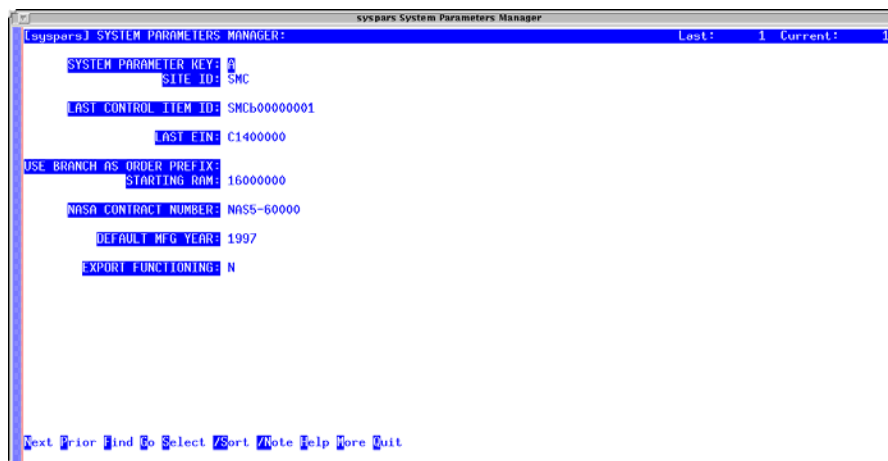
Field Name	Data Type	Size	Description
EMPLOYEE NUMBER (EMPL)	String	10	Unique identifier for an employee.
LAST NAME (EMPLOYEE)	String	30	Last name of the employee.
FIRST NAME (EMPLOYEE)	String	30	First name of the employee.
STATUS (EMPLOYEE)	String	1	Status of the employee.
WORK CENTER (EMPLOYEE)	String	6	Code for work center where the employee is normally assigned.
PHONE (EMPLOYEE)	String	18	Telephone number of the employee.
FAX NUMBER (EMPLOYEE)	String	13	FAX number of the employee.
E-MAIL (EMPLOYEE)	String	30	E-mail address for the employee.
PAGER NUMBER (EMPLOYEE)	String	13	Pager number for the employee.
CC MAIL (EMPLOYEE)	String	30	CC-mail address of the employee.

27.12.2 System Parameters Manager Screen

The System Parameters Manager screen (Figure 27.12.2-1) is for maintaining system-wide XRP-II parameters and is principally used when first installing the system. Since ILM uses only a subset of the full XRP-II capabilities, this is a scaled down version of the screen described in the Section 6 of the *XRP-II System Reference Manual*. It contains only the fields needed to tailor the system to the site at which it operates.

Several fields have particular significance for ILM. The Site ID field contains the code for the ECS site where the operator's copy of XRP-II is installed. The field is interrogated by ILM processes that have to determine which assets belong to the local site. The Last EIN field is used by XRP-II to keep track of the most recently used, automatically assigned EIN. It updates the field whenever an operator presses <RETURN> in the EIN field when creating records via EIN Entry. The NASA Contract Number and Default MFG Year fields contain values used as defaults when creating ILM records, and the Export Functioning field precludes more than one export process from running at a time because they would conflict.

Table 27.12.2-1 describes each of the screen's fields.



The screenshot shows a window titled "syspars System Parameters Manager". The window has a blue header bar with the title and "Last: 1 Current: 1". The main area contains several fields with labels and values:

- SYSTEM PARAMETER KEY: A
- SITE ID: SMC
- LAST CONTROL ITEM ID: SMCb00000001
- LAST EIN: C1400000
- USE BRANCH AS ORDER PREFIX: Y
- STARTING RAM: 16000000
- NASA CONTRACT NUMBER: NAS5-60000
- DEFAULT MFG YEAR: 1997
- EXPORT FUNCTIONING: N

At the bottom of the window, there is a menu bar with the following options: Text, Prior, Find, Go, Select, Sort, Note, Help, More, and Quit.

Figure 27.12.2-1. System Parameters Manager

Table 27.12.2-1. System Parameters Manager Field Descriptions

Field Name	Data Type	Size	Description
SYSTEM PARAMETER KEY	String	1	Code that designates the active record in XRP-II's system parameter table. The active record must have the value "A".
SITE ID	String	6	Code that identifies the ECS site where this XRP-II system is installed.
LAST CONTROL ITEM ID	String	20	Code used in determining the next sequentially available identifier when assigning control item identifiers automatically.
LAST EIN	String	20	Code used in determining the next sequentially-available identifier when assigning EIN numbers automatically
USE BRANCH AS ORDER PREFIX	String	1	Code that, if "Y", causes all new purchase orders, work orders, and sale orders to be prefixed with the site code of the operator or, if null, the default site code.
starting RAM	Number	8	Initial amount of memory XRP-II is to use
NASA CONTRACT NUMBER	String	11	Code that is used by NASA to identify the ECS contract. It is attached to all property records
DEFAULT MFG YEAR	String	4	Year used as default to identify when an item was built.
EXPORT FUNCTIONING	String	1	Code that indicates if an XRP-II data "export" function is in progress; used to prevent multiple export routines being run concurrently

27.12.3 Inventory Location Manager Screens

This screen (see Figure 27.12.3-1 and Table 27.12.3-1) provides for the maintenance of location information used in the inventory and logistics processes.

The screenshot displays the 'Inventory Location Manager' window. The title bar reads 'Imlocms Inventory Location Manager'. The window content shows a form with the following fields and values:

- MATERIAL LOCATION ID: EDC
- LOC DESC: EROS Data Center
- LOCATION TYPE: W (Null or S=stock R=received N=non-nettable W=work center A=archive)
- SITE: EDC
- Shipping Report Number: 1
- Shipping Report Alpha: K
- CONSIGNEE NAME: EROS Data Center
- ADDRESS 1: Mundt Federal Building
- ADDRESS 2: Receiving Warehouse
- CITY: Sioux Falls
- STATE: SD
- ZIP: 57198
- PHONE: (605)594-6818

At the bottom of the window, there is a menu bar with the following options: Text, Prior, View, Find, Go, Select, Sort, Note, Help, More, and Quit.

Figure 27.12.3-1. Inventory Location Manager Screen

Table 27.12.3-1. Inventory Location Manager Field Descriptions

Field Name	Data Type	Size	Description
MATERIAL LOCATION ID	String	6	ID for the location where material can be found.
DESCRIPTION (Material Location)	String	30	Text description of the utility of the site.
LOCATION TYPE (Material Location)	String	1	Code that specifies the material application at the site: Null or S = stock, R = received, N = non-nettable, W = work center, A = archive.
SITE (LOCATION)	String	6	Code for the ECS site hosting the inventory location.
SHIPPING REPORT NUMBER	Number	2	Report number assigned to this item when the item was shipped.
SHIPPING REPORT ALPHA	String	2	Shipping report code associating an alpha code to a numeric site code. See Shipment Numbers by Site screen (shipno).
CONSIGNEE NAME	String	30	Name of individual/office responsible for material at the site.
ADDRESS 1 (Consignee)	String	30	First part of address
ADDRESS 2 (Consignee)	String	30	Second part of address.
CITY	String	20	City part of address
STATE	String	2	State 2 character abbreviation of address.
ZIP	String	10	Zip code of address.
PHONE	String	18	Telephone number of address

27.12.4 Hardware/Software Codes Screens

This screen (see Figure 27.12.4-1 and Table 27.12.4-1) provides the maintenance of the codes used to identify maintenance cost source information in the inventory and logistics processes.

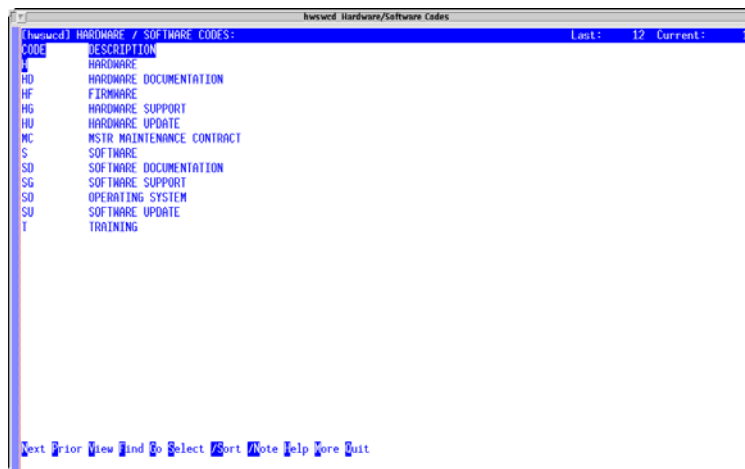


Figure 27.12.4-1. Hardware/Software Codes Screen

Table 27.12.4-1. Hardware/Software Codes Field Descriptions

Field Name	Data Type	Size	Description
CODE (Hardware/Software)	String	10	Code for classifying the type of items.
DESCRIPTION (Hardware/Software)	String	30	Description for the hardware/software code

27.12.5 Status Code Manager Screen

The Status Code Manager screen (Figure 27.12.5-1) maintains a set of standardized status codes for tracking property and events in the inventory and logistics processes. Table 27.12.5.1 describes the screen's fields.



Figure 27.12.5-1. Status Code Manager

Table 27.12.5-1. Status Code Manager Field Descriptions

Field Name	Data Type	Size	Description
CODE	String	4	Code for an inventory status for an item.
DESCRIPTION	String	30	Description for the code.

27.12.6 DAAC Export Inventory Data Screen

ILM at the SMC can maintain consolidated records about inventory, logistics, and maintenance activities system-wide. Records created at local sites can be exported and shipped to the SMC where they can be added to records that were centrally created. For ECS, only records about items at the site are to be exported.

The DAAC Export Inventory Data utility supports this customized export process. It generates a formatted data file containing site records changed but not previously exported, and optionally

transfers the file via ftp to a machine at the SMC. Operators at the SMC use the ILM Import Records utility (see Section 27.12.7) to load the data into the system there.

The screen in Figure 27.12.6-1 initiates the export process. XRP-II analyzes the transaction log to determine what data changed since the last time the function was used and which site items were affected. EIN, EIN structure, purchase order, work order, inventory and transaction history records are copied and stored in files compatible with XRP-II's ILM Import Records utility. These files are, in turn, archived in a tar file. The tar file is given a name that identifies the date and time the export was done, the origination site, the file's type, and the machine to which the file is to be sent. If the SEND NOW feature is used, XRP-II transfers the files via ftp then moves them from the export directory to an archive directory. Otherwise, the files remain in the export directory to be transferred manually.

NOTE: ILM import and export are being done manually daily through cronjobs at all the sites. The cron jobs perform the same tasks as described in Sections 27.12.6 and 27.12.7.

Note: Export files that are transferred manually to a destination machine must also be moved manually to the export archive directory.

Note: The export directory and its corresponding export archive directory are configuration parameters named via program environment variables set in the XRP-II configuration files during installation.

Enter the name of the machine to receive the data (using its domain name or IP address), and choose whether or not to ftp the tar file immediately after it is created. The name can be selected from a managed list by using XRP-II's /Zoom command. Use Execute to begin data extraction and, if prompted, provide a login account and a password for the ftp. As processing progresses, XRP-II will display informational messages, including some that contain the name of the tar file that are created. Messages that terminate with the symbol ">" require an operator response. Hit any key and processing will continue. XRP-II returns to the System Utilities menu when done.

Table 27.12.6-1 describes the screen's fields.

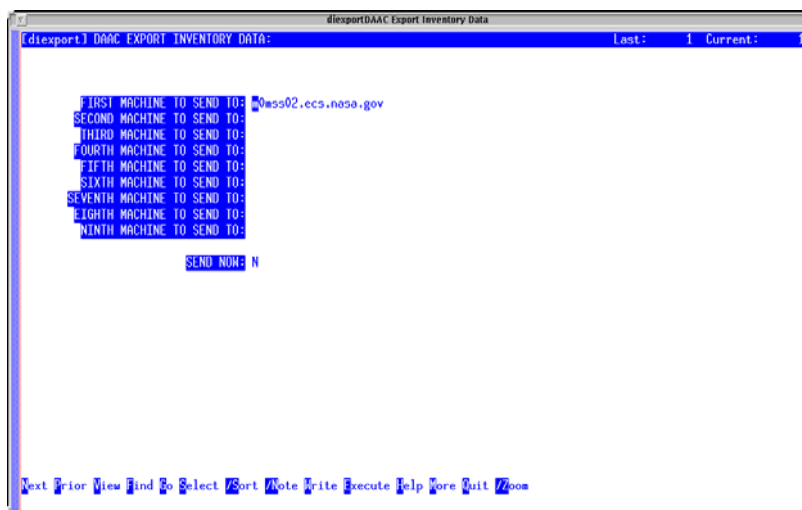


Figure 27.12.6-1. DAAC Export Inventory

Table 27.12.6-1. DAAC Export Inventory Data Field Descriptions

Field Name	Data Type	Size	Description
FIRST MACHINE TO SEND TO	String	40	Full domain name or IP address of the machine to receive the exported inventory data.
SEND NOW	String	1	Flag to indicate if the export tar file is to be sent now.

27.12.7 ILM Import Records Screen

ILM data will be exchanged among ECS sites on a routine basis. The ILM Import Records utility is designed to load data from tar files that had been created and forwarded using either of XRP-II's ILM data export utilities (see Sections 27.12.6).

The screen shown in Figure 27.12.7-1 initiates the import process. Entering “Y” at the prompt causes XRP-II to process all files in the directory named in the IMPORTPATH environment variable. Import tar files -- whose names indicate the date and time they were made -- are processed in chronological order as determined from their file names. Upon completion, the original files are moved to an archive directory named in the IMPORTARC environment variable.

NOTE: ILM import and export are being done manually daily through cronjobs at all the sites. The cron jobs perform the same tasks as described in Sections 27.12.6 and 27.12.7.

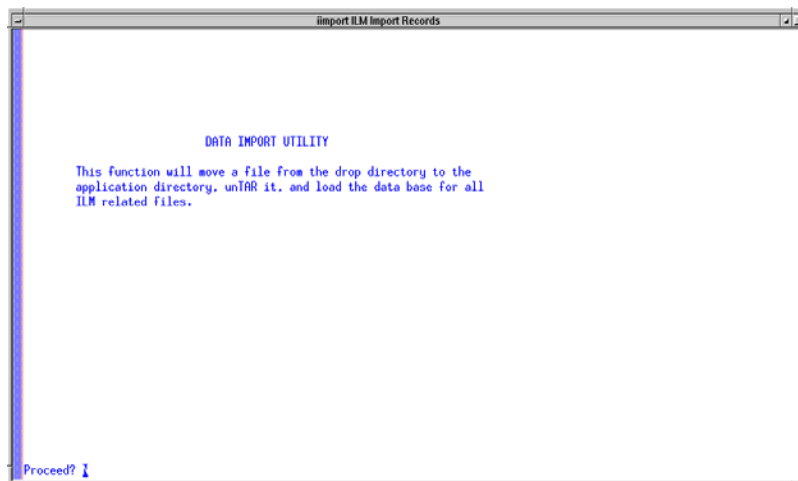


Figure 27.12.7-1. ILM Import Records

27.13 System Utilities Menu (Perform XRP-II Master File Maintenance)

XRP-II groups together several programs that help standardize values for all of XRP-II and support inter-site exchange of Baseline Manager data. The screens supporting these programs are accessed via the System Utilities menu (Figure 27.13-1 and Table 27.13-1).

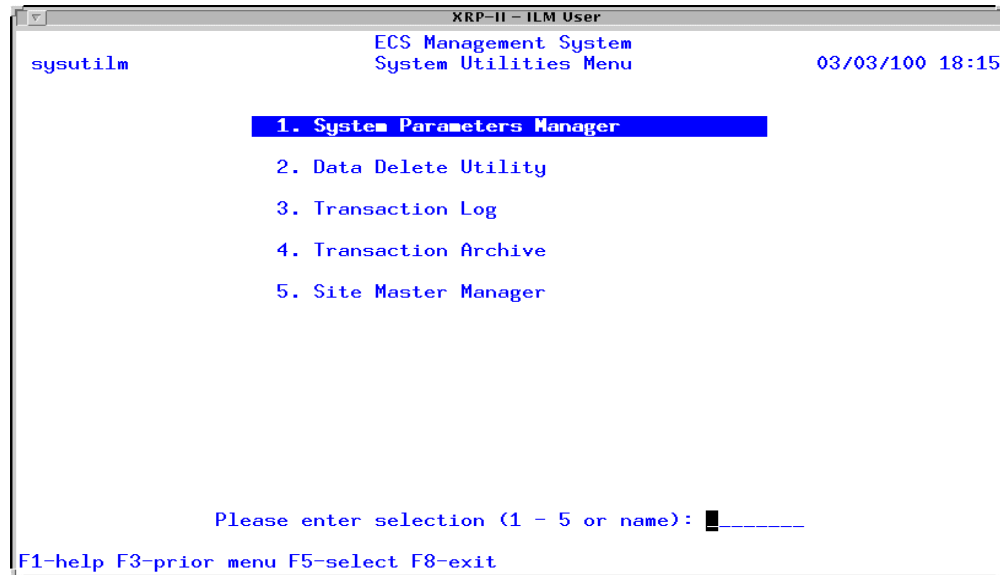


Figure 27.13-1. System Utilities Menu

Table 27.13-1. System Utilities Options

Menu item	Function	Section
System Parameters Manager	Contains fields that needed to tailor the system at a site.	Refer to section 27.12.2
Data Delete Utility	Enables the deletion of data that is no longer wanted in the database	27.13.2
Transaction Log	Contains all the transaction log of all the transaction user performs on the system.	27.13.3
Transaction Archive	Enables the archiving of all the old transaction logs.	27.13.4
Site Master Manager	Maintains information about all the ECS-related sites.	27.13.5

27.13.1 Data Delete Utility

Data Delete Utility (see Figure 27.13-1) enables the deletion of arbitrary types of data that are no longer wanted in the database (see Table 27.13-1 for procedure). Access to this screen should be

restricted to the System administrator. In a relational database with referential integrity it is not possible to delete a record in a database table that is referenced by a record in some other table. Data delete utility enables the user to specify a table and key value, then it identifies on the screen the data related to the specified data, and optionally allows the user to delete the related data and the base record.

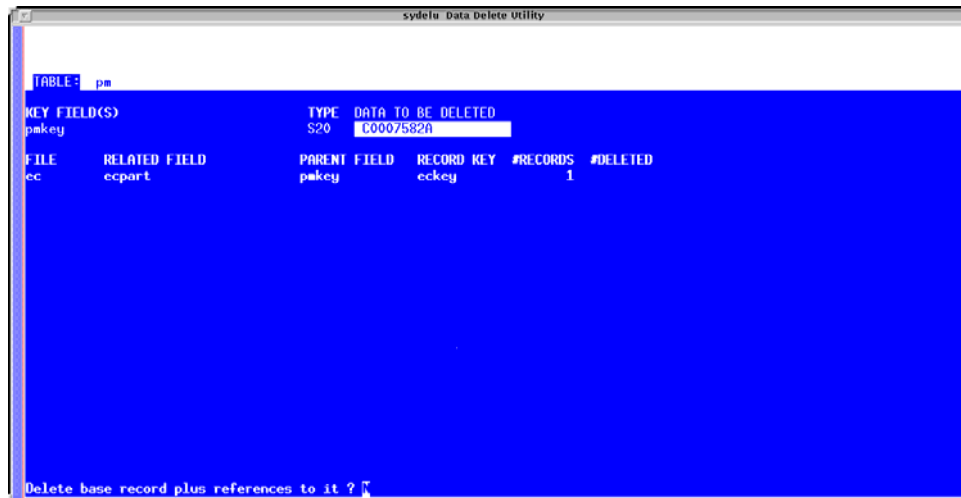


Figure 27.13.1-1. Data Delete Utility

Table 27.13.1-1. Procedure to Delete Base Records in the Database

PERFORM	ACTION
Navigate to the Data Delete Utility screen	From the Main Menu A. Select System Utilities Menu – press ‘enter’ B. Select Data Delete Utility – press ‘enter’
Enter data to be deleted	A. Enter the table name of a database table at the “Table:” prompt. XRP responds by listing the one or more fields which form the key of the file. B. Then enter the key value identifying the base record to be deleted. C. XRP specifies the number of record selected based on the key value the user entered. Press ‘enter’ to go to the next step. D. XRP prompts “ DELETE MODE: Confirm Silent or Quit? ” Confirm – Type ‘C’ to select confirm. XRP identifies and lists on the screen all of the tables in the database have one or more records referencing the base record. If the user confirms that the data is to be deleted, XRP deletes all of the related records and then the base record. The number of records deleted from each table is echoed to the screen. Silent – XRP will delete all the base records silently without asking the user to confirm the deletion. Quit – to exit out of the delete mode.
Exit the screen	Press ‘F3’ to exit Data Delete Utility screen.

27.13.2 Transaction Log Screen

The screen shown in Figure 27.13.2-1 allows operators to browse, and maintain if necessary, the database transaction log file. Values for all fields on this screen are system-supplied. When a database record is modified via a data entry screen, the system provides the next available transaction number and record information about what field was modified when and by whom.

The transaction log facilitates synchronizing database changes among sites. For example, the Export Site-Unique Changes utility (Section 27.12-6) analyzes the log’s entries to identify database records that have been modified, setting each Transferred field so it bypasses the entry next time the utility is run.

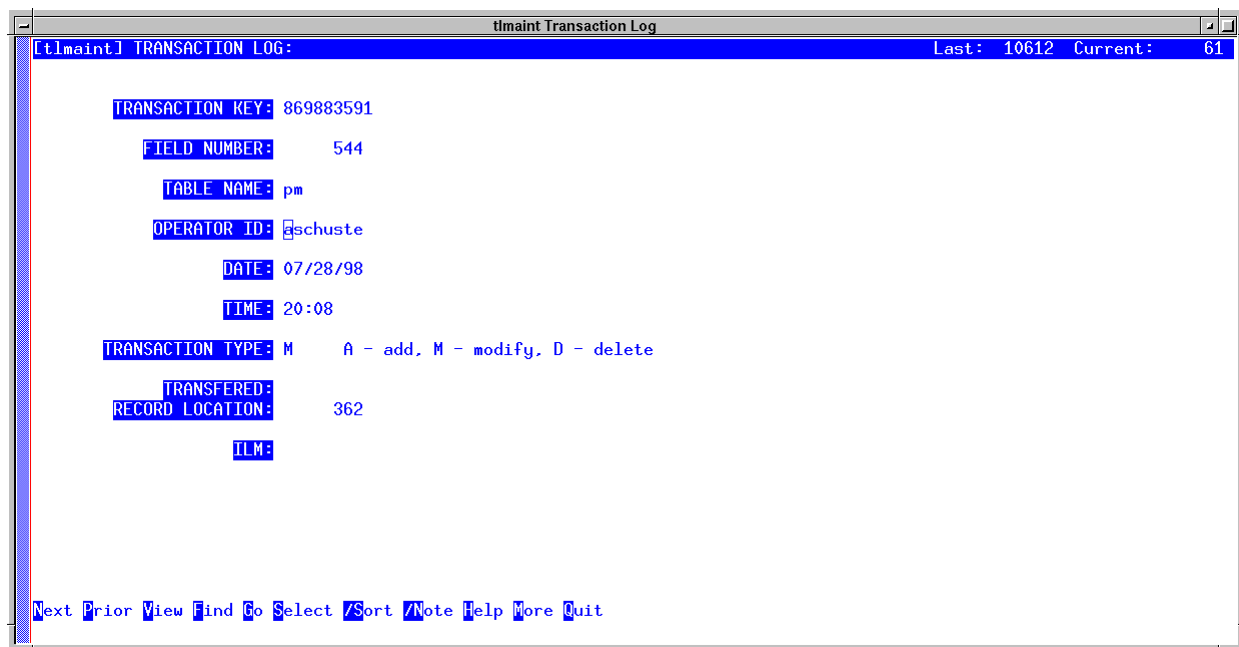


Figure 27.13.2-1. Transaction Log

The screen displays numbers to identify XRP-II database fields because field names are not stored in the database. Field names that correspond to the numbers can be found in file \$MSPATH/mms/def/file.h, where MSPATH is an environment variable identifying XRP-II's installation directory.

Similarly, the screen uses numbers to identify locations of altered database records. The current contents of a referenced record can be displayed as follows, but only if the record at that location was neither deleted nor replaced by another since the log entry was made. At a Unix command line prompt, type:

SYS920	Invokes UNIFY's database test driver
setloc <table> <location>	Displays a record's data
end	Exits program SYS920

Note: Use Transaction Archive (Section 27.13.4) to remove obsolete transaction records. Transaction Archive preserves records that export utilities still need, and it saves a historical copy of the records it deletes.

Table 27.13.2-1 describes this screen's fields.

Table 27.13.2-1. Transaction Log Field Descriptions

Field Name	Data Type	Size	Description
transaction key	numeric	5	Number that uniquely identifies each update transaction
field number	numeric	8	Numerical identifier for the XRP-II field affected by the transaction.
table name	string	10	Name of the XRP-II table affected by the update transaction
operator id	string	8	Userid of the operator making the update transaction
date	date	N/A	Date of the update transaction
time	time	N/A	Time of the update transaction
transaction type	string	1	Code for the type of transaction: A (add), M (modify), or D (delete)
transferred	string	1	Code that indicates that the transaction has been analyzed by an export utility. "T" means the corresponding control item record has been exported, while "X" means it did not need to be exported.
record location	numeric	8	Identifier for the relative record within the XRP-II table affected by the update transaction
ilm	string	1	Code that distinguishes between ILM-and BLM-related log entries; "Y" signifies ILM

27.13.3 Transaction Archive Screen

Control item data update transactions should periodically be deleted from the database after changed records have been exported. This makes room to log new transactions.

The Transaction Archive screen shown in Figure 27.13.3-1 copies to a named file the records of transactions that occurred on or prior to a specified cutoff date. It then deletes the records from the database.

Specify the date of the last transaction to archive and the name of a file in which to store the data.

Note: XRP-II will only archive a transaction log record if its Transferred field contains the value "T" or "X". The presence of a "T" or "X" means the record has been analyzed by a program for exporting records about control item changes to other sites (see Section 27.13.2). Deleting unanalyzed transaction log records can cause incomplete data exchanges.

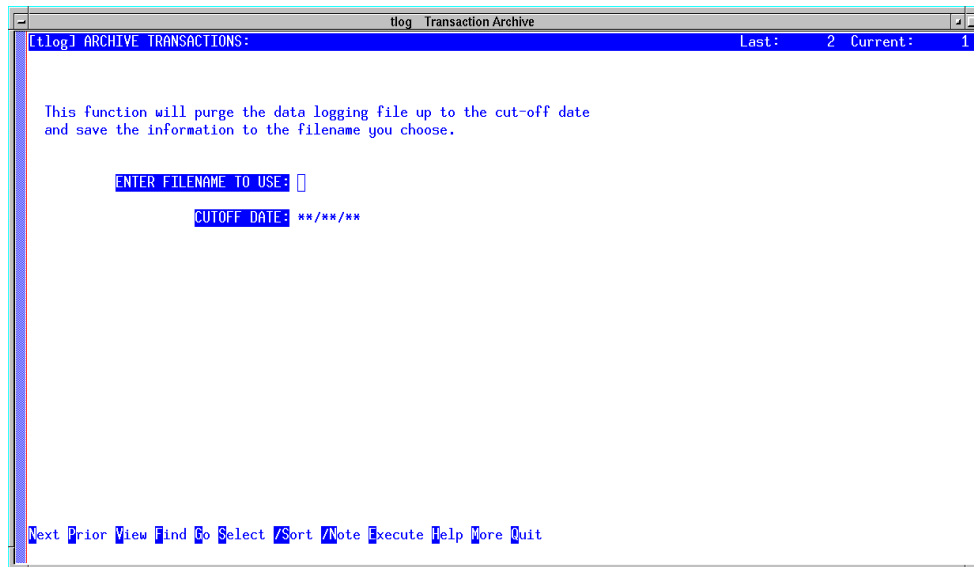


Figure 27.13.3-1. Transaction Archive

Table 27.13.3-1. Transaction Archive Field Descriptions

Field Name	Data Type	Size	Description
file name	string	8	Name of the file in which to store transaction records being archived
cutoff date	date	N/A	Date of the most recent transaction to be archived

27.13.4 Site Master Manager Screen

The Site Master Manager screen (Figure 27.13.4-1) lets operators maintain an index of ECS-related sites. This index, which identifies details about each site, also serves as a pick list to facilitate entering control item implementation status data and report generation parameters on other screens.

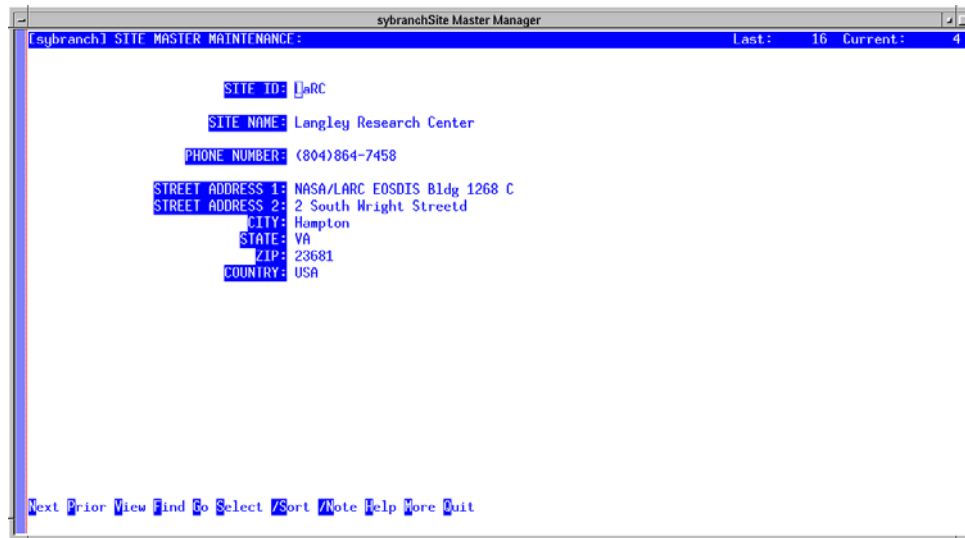


Figure 27.13.4-1. Site Master Manager

Table 27.13.4-1 describes the “site” fields that were tailored for ECS.

Table 27.13.4-1. Site Master Manager Field Descriptions

Field Name	Data Type	Size	Description
site id	string	6	Code that uniquely identifies an ECS site.
site name	string	46	Full name of an ECS site.
phone number	string	8	Phone number of a point of contact at the site
street address	string	30	Address for the site
city	string	20	Name of the city in which the site is located
state	string	2	Abbreviation for the state in which the site is located
zip	string	10	Postal code for the site
country	string	16	Code for the country in which the site is located

27.14 System Tools Menu (Perform XRP-II System and Database Administration)

XRP-II provides several programs for controlling user access, maintaining the database, and customizing the user interface. Many of the programs are supported by data entry screens accessed via XRP-II's System Tools menu (Figure 27.14-1 and Table 27.14-1).

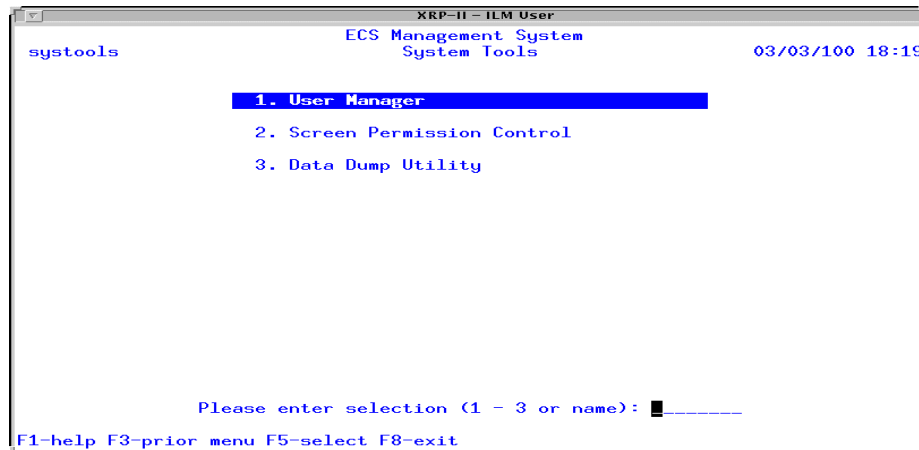


Figure 27.14-1. System Tools

Table 27.14-1. System Tools Options

Menu item	Function	Section
User Manager	Use to register users to XRP.	27.14.1
Screen Permission Control	Let operator specifies the XRP-II menus and data entry screens a user or group can access.	27.14.2
Data Dump Utility	To bulk dump one or more XRP-II database tables into specially formatted data files.	27.14.3

27.14.1 User Manager Screen

Operators use the User Manager screen (Figure 27.14.1-1 and Table 27.14.1-1) to register the Unix userids of individuals authorized to run XRP-II. Individuals are assigned a group of menus and screens that may be accessed and a specific entry menu. As part of logon processing, XRP-II's menu handler obtains an individual's Unix userid from the system and verifies it against those that have been registered.

Use this screen to add, delete, modify or browse XRP-II user records.

Note: Data access for operators running XRP-II executables from the command line is governed by Unix file permissions rather than the menu handler.

The screenshot shows a window titled 'users User Manager'. The top bar indicates 'Last: 27' and 'Current: 1'. The main area displays the following fields and values:

- LOGIN ID: ttran
- PASSWORD: ttran
- FIRST NAME: Trang
- LAST NAME: Tran
- TITLE:
- TYPE:
- GROUP: ilmadmnd
- SITE:
- ADDRESS:
- CITY:
- PHONE:
- STATE:
- ZIP:
- START MENU: mainm

At the bottom, there is a menu bar with options: Text, Prior, View, Find, Go, Select, Sort, Note, Help, More, Quit.

Figure 27.14.1-1. User Manager

Table 27.14.1-1. User Manager Field Descriptions

Field Name	Data Type	Size	Description
login id	String	32	Full, network-addressable name of a host
password	String	6	Code that uniquely identifies an ECS site
first name	String	30	First name of the user
last name	String	30	Surname of the user
title	String	20	Name of the user's position or job
group	String	8	Name for a collection of XRP-II data entry screens and menus. These are the default screens and menus the user can access. Deviations can be specified via Screen Permission Control.
address	String	30	Street address where the responsible engineer is located
city	String	20	Name of the city in which the responsible engineer is located
phone	String	18	Phone number for the responsible engineer
state	String	2	Name of the state in which the responsible engineer is located
zip	String	10	Postal code where the responsible engineer is located
start menu	String	8	Name or code of the menu initially presented to the user at login

27.14.2 Screen Permission Control Screen

Screen Permission Control (Figure 27.14.2-1) lets operators specify the XRP-II menus and data entry screens a user or user group can access and the data manipulation permissions the user or group is granted when accessing a screen. It replaces reliance on the “users” and “groups” files

discussed in Sections 5 and D.5 of the *XRP-II System Reference Manual*, however the concept of access and privileges by group and user is the same.

Use this screen to browse, add to, or edit existing screen permission control records. Each record renders a menu or data entry screen accessible to some user or group. For data entry screens, it also assigns to the user or group the privilege to query (inquire), add, modify, and/or delete records via the screen. A privilege is assigned by placing a “Y” in the appropriate privilege field. Similarly, a privilege is removed by placing an “N” in the appropriate privilege field.

Consider the following when modifying screen permissions:

- Privileges specified for a user take precedence over privileges specified for the user’s group
- A user or group is granted access to a menu or screen only if a privilege is assigned.
- Assignment of All Privilege overrides other privileges specified in the record
- Privileges specified in the record do not override permission restrictions coded into specifications for the screen (e.g., no user can update the database via a screen marked for querying the database only, regardless the privileges the user is given for the screen.)

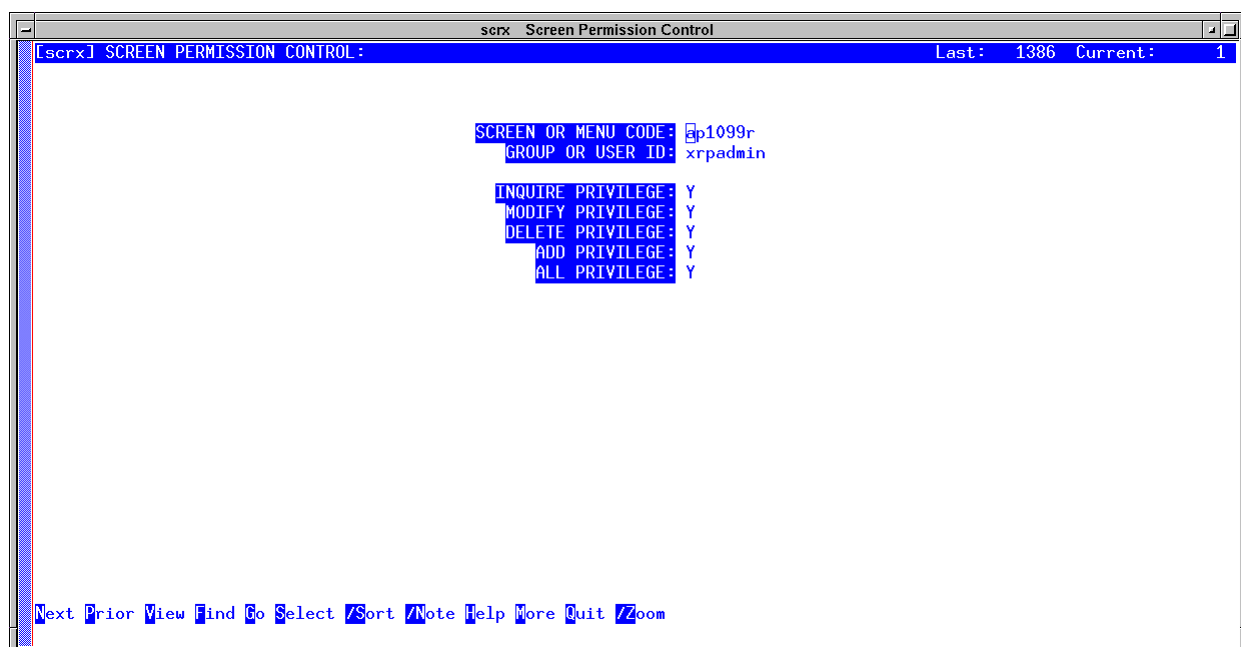


Figure 27.14.2-1. Screen Permission Control

ILM is delivered with a default set of user groups and associated screen permissions (see Section 27.2). Example ways an operator might extend this set include:

- Making a screen available to a group - add a record that names the screen and group, then assign at least one privilege for the group
- Making a screen accessible to only certain users - add a record for each user. Name the screen and the user, and assign at least one privilege for the user. Then find and delete any records that make the screen accessible to groups other than any established for the system administrator.
- Increasing a screen's privileges for a member of a group - add a record that names the screen and the user, and assign the extra privilege for the user.
- Restricting a screen's privileges for a member of a group - add a record that names the screen and the user, and remove the privilege(s) for the user

Table 27.14.2-1 describes this screen's fields.

Table 27.14.2-1. Screen Permission Control Field Descriptions

Field Name	Data Type	Size	Description
screen or menu code	string	8	Identifier (short name) that XRP-II programs use in referencing an XRP-II screen or menu (e.g., pici).
group or user id	string	8	Name that classifies XRP-II users according to access required to menus and screens, or an individual's Unix userid
inquire privilege	string	1	Code indicating if the group or user can: a) read database records via the named screen; or b) can access the named menu.
modify privilege	string	1	Code indicating if the group or user can: a) modify database records via the named screen; or b) can access the named menu.
delete privilege	string	1	Code indicating if the group or user can: a) delete database records via the named screen; or b) can access the named menu.
Add privilege	string	1	Code indicating if the group or user can: a) add database records via the named screen; or b) can access the named menu.
All privilege	string	1	Code indicating if the group or user can: a) read, modify, delete, and add database records via the named screen; or, b) can access the named menu.

27.14.3 Data Dump Utility Screen

Operators use the Data Dump Utility screen (Figure 27.14.3-1 and Table 27.14.3-1) to bulk dump one or more XRP-II database tables into specially formatted data files. A file is created for each database table, and it contains all fields for all records in the table. Fields are separated by pipe symbols (|). The first line in each file identifies the field ordering. See the XRP Tools,

Techniques, and Conventions Manual, Sections 1.5 and 1.6, for file format conventions XRP-II uses.

Enter Modify mode and specify which tables to dump and whether to archive the resulting data files. Return to Inquiry mode, then enter “E” to initiate the dump and “Y” at the confirmation prompt. XRP-II will create the requested data files and return to the System Tools menu.

Note: If a tar file is named, XRP-II archives all formatted data files it finds regardless whether the corresponding database table was part of the current dump.

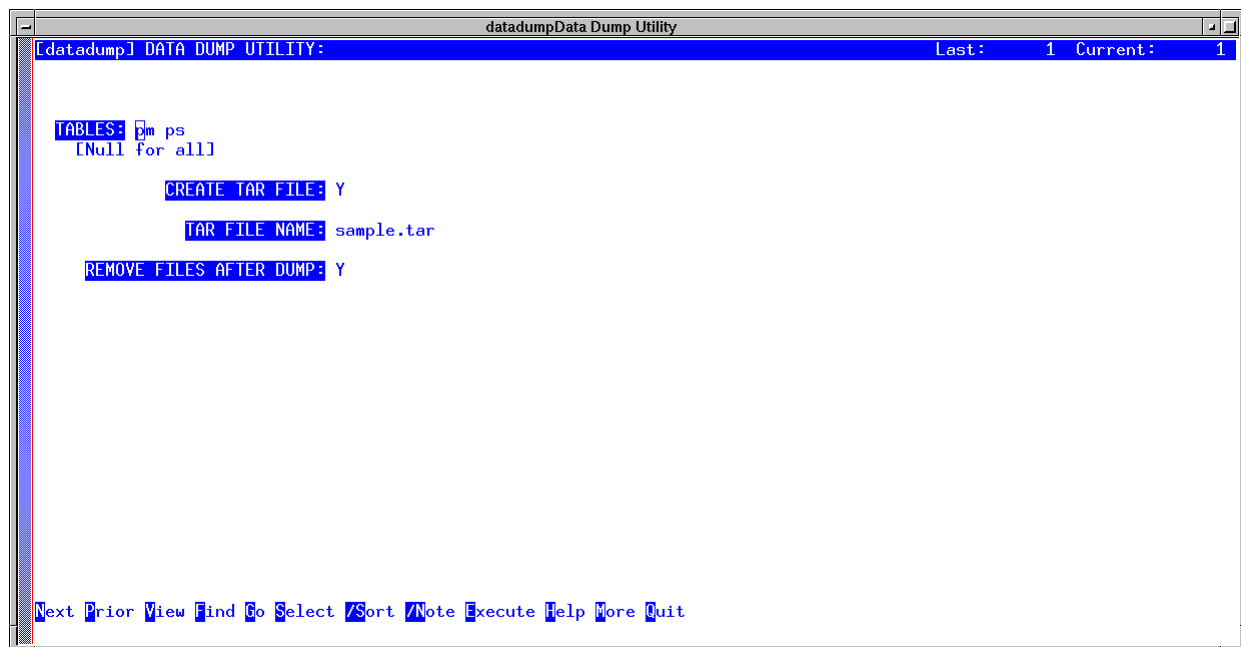


Figure 27.14.3-1. Data Dump Utility

Table 27.14.3-1. Data Dump Utility Field Descriptions

Field Name	Data Type	Size	Description
dump	string	60	List of tables to dump (e.g., pm, ec, etc.). A null field causes all tables to be dumped.
tar dump files	string	1	Code indicating whether or not to tar the data tables that were dumped
tar file name	string	40	Pathname for the tar file in, or relative to, the database directory specified in environment variable \$DBPATH

28. Maintenance of Configuration Parameters

28.1 Purpose

These procedures describe the overall maintenance of the ECS Configuration Parameters Baseline for ECS custom software and hardware, including patches, database, operating systems, Commercial-off-the-shelf (COTS) and networks.

The ECS Configuration Parameter Baseline comprises of capturing, controlling, managing, documenting and verifying updates to the ECS Configuration Parameter Baseline at the ECS Development Facility (EDF). It also sets the direction for EDF to capture and document approved changes made by the Data Active Archive Center (DAAC) personnel to their operational configuration parameter baseline. The only assumption is that an initial ECS configuration parameter baseline has been established through the appropriate ECS Configuration Change Request (CCR) and ECS Change Control Board (CCB) approval process.

28.2 Applicability/Scope

This document is applicable to all organizational activities within the ECS Development Facility (EDF) and DAAC(s) that result in a change to the ECS Configuration Parameter Baseline.

This document adheres to the policies, guidelines and procedures of the following ECS Plan, ECS Baseline Technical Document, Project Instructions (PI) and WIs.

102-CD-003	Configuration Management Plan for the Science Data Processing Segment of the ECS Project
910-TDA-022	Custom Code Configuration Parameters
PI CM-1-004	CCB Change Control Process
PI MO-1-003	ECS SDPS Sustaining Engineering and Maintenance Management
WI MO-1-003-4	ECS Deployment IPT Work Instruction
WI MO-1-003-5	Operations Class NCR Management Process

28.3 Major Stakeholders and Supporting Organizations

The Maintenance and Operations, Development Engineering, Systems Engineering, Sustaining Engineering, Program Management, Quality Office, Testing, Deployment, DAAC(s) and ESDIS organizations are the primary stakeholders in this process.

28.4 Definitions

Configuration Parameter Change - A configuration parameter change as defined in this document, refers to any change that results in the modification of an existing configuration parameter, addition of a new configuration parameter, or deletion of an existing configuration parameter. A configuration parameter change can be a result of ECS updates to the ECS custom software/hardware, patches, database, operating systems COTS and networks, or due to altering of a parameter setting or a value range of an existing configuration parameter.

Configuration Change Request (CCR) - A document that requests, defines and justifies a change(s) to design and/or documentation which is controlled by a CCB. In the context of the configuration parameter, a CCR is required to propose a change to the ECS Configuration Parameter Baseline. A CCR written against a Configuration Parameter Baseline will follow the normal path of CCR process, review, and appropriate CCB approvals.

ECS Change Control Board (CCB) - ESDIS CM consists of hierarchy of boards that control the configuration. These consist of ESDIS CCB (top level), ECS CCB (middle level), lower level CCBs, such as, Maintenance and Operations (M&O) CCB, Science Development (SD) CCB and ECS Development Facility (EDF) CCB, and the lowest level CCBs at the DAACs.

COTS - Commercial off-the-shelf (COTS) hardware platforms, or software products. These tools may be brought into the EDF either permanently (e.g., as an approved software development tool) or temporarily (e.g., for the purpose of testing and/or evaluation).

DAAC - Distributed Active Archive Center is one of several operational sites where science data is ingested and processing is performed within EOSDIS.

DDTS - Distributed Defect Tracking System is a UNIX change management and bug tracking system that tracks and manages defects (Non-Conformance Reports - NCRs) throughout the software lifecycle.

EDF - ECS Development Facility, Upper Marlboro, MD; also known as Landover.

EDF/SCDV CCB - EDF/SCDV CCB is responsible for establishing and managing ECS Baseline development. This includes controlling development configurations; recommending EDF procurement to the ECS CCB; approving all non-procurement internal CCRs to change the EDF configuration, COTS, hardware/software, networks, facilities and related procedures; Controlling all EDF documents, and installation /removal of all evaluation COTS in the EDF. EDF/SCDV also has the responsibility for controlling ECS releases. It controls the configurations at the test environment (PVC and VATC), and recommends test environment procurements within the contractual baseline. It authorizes segment turnovers and toolkit distribution and migration. It approves release-specific CDRL documents, Class II release-specific changes, white papers and technical documents, ECS Baseline Technical Documents, and distribution of ECS software and hardware outside of the EDF for use within the ECS project prior to CSR.

M&O CCB - Maintenance and Operations Configuration Control Board, is responsible for agenda items on providing technical support to operational sites and SMC, and is responsible for M&O related CCRs and NCRs written at the DAAC(s) that are reviewed for closure. M&O is

the approving authority on any hardware or software configuration changes to the system operational baseline. Any ECS configuration baseline changes to the DAAC(s) and SMC operational baseline, must be approved by the M&O CCB first.

Non-Conformance Report (NCR) - The description of a problem discovered at the EDF within the development, PVC or VATC environments, or at the DAAC(s) and SMC. NCRs are managed using the DDTS tool.

Operational Site - One of the DAAC(s) or the SMC.

Patch - Package of data/executables which may require more complex installation than simply copying into the operational system. Used to deliver fully tested/verified fixes for OPS Priority List NCRs (and other fixes that have been implemented in the code being delivered. Patches may also be used to deliver new functionality and/or customer-directed enhancements that require delivery prior to the next scheduled Software Release. Pre-release delivery may be justified by the need to support customer program priorities such as the scheduled testing of ECS external interfaces or planned availability of new production data.

Problem Review Board - Daily review board is responsible for the review and approval of NCRs entered into the REL_B0_INFORMAL 1 DDTS class, and in the OPERATIONS class that are ready for closure.

PVC - Performance Verification Center - A simulated DAAC operational environment within the EDF, used for testing and evaluating performance of ECS custom software, databases, COTS, operating systems and networks.

SMC - System Monitoring Center. The coordination and monitoring node of the operational portions of the ECS SDPS system.

Sustaining Engineering Effort - Responsible for implementing needed improvements to the current operational version of the hardware, software, and firmware. The sustaining engineering function includes analysis, identification of processes, procedures and methods to accommodate the following: new technologies and concepts; manage system upgrades and evolution; control and maintenance of ECS databases; and activities necessary to ensure ECS reliability, maintainability, and availability. The sustaining engineering effort also includes the development, test, installation, configuration, and tuning of the ECS software, COTS packages, operating systems, compilers, tools, utilities, networks and databases.

Test Executable (TE) - Package of data/executables that are easily installed into the operational system, and extracted just as easily. Used to deliver a fix for an emergency problem. The fix must be merged into the maintenance baseline and tested by the developers under oversight by Software Integration and Test (SWIT).

Trouble Ticket (TT) - A description of a problem, or an unresolved operational issue, discovered at one or more DAAC(s) or the SMC. TTs are documented using the Remedy Tool. Unresolved Trouble Ticket can be upgraded to an NCR level.

VATC - Verification and Acceptance Test Center - Test environment located within the EDF.

28.5 Procedure/Responsibilities

The procedures in this document for Maintenance of Configuration Parameters Baseline, are in close compliance with plans, PIs and WIs referenced in this document. The author will not discuss any information that is already documented in the referenced documents. Reader is advised to review the appropriate references (documents listed in sections 3.2 and 6.1.1) for details on topics covered in this WI.

28.5.1 Procedure Documentation

For details on specific roles and responsibilities of CCBs, and associated process and procedures, some related documents are as follows: ESDIS Change Control Process (documents 432-10-21, *ESDIS CM Procedures*, and 423-10-21-1, *Science Systems PCMB CM Procedure*). ECS CCB procedures are described in PI CM-1-004, *CCB Change Control Process*, and document control is described in DM-1-001, *Documentation Management and Control*. The M&O Change Control Process can be found in PI MO-1-003, *ECS SDPS Sustaining Engineering and Maintenance Management*, and Sustaining Engineering and Maintenance Management is described in WI MO-1-003-4, *ECS Deployment IPT Work Instruction*.

Roles of lower-level CCBs are outlined in the 'definitions' sections. The procedures to change the Configuration Parameter Baseline will follow the current CCR and CCB approved processes and guidelines. CCRs falling under the authority of lower-level CCBs, and requesting a change to DAAC operational Configuration Parameter Baseline shall be approved by the ECS M&O CCB, in addition to lowest-level DAAC CCBs and Board approvals. ECS M&O will make the final decision on any changes implemented to the DAAC and SMC Operational Configuration Parameter Baseline. For further details refer to PI CM-1-004.

28.5.2 Parameter Change Control Procedure

Configuration parameters are 'controlled' by either DAAC or the EDF. Proposed changes to these configuration parameters will originate from the controller of these parameters, in most cases. The one exception is when new ECS software/hardware patches or versions warrant new configuration parameters, or changes to the existing parameters. In cases, where one of the entities, i.e., DAAC or Landover, proposes a change to an existing configuration parameter which it 'does not' control, then the requesting entity will follow the appropriate ECS change request, resolution and CCB approval process of the entity that controls that parameter.

Configuration Parameter Baseline documents defines information on various areas of ECS. These areas are, but are not limited to the following: ECS Custom Code configuration, program and application configuration files and parameters; Sybase Server Databases; Operating System - build, options of auto mount, and Kernel configuration files and parameters; and COTS configuration files and parameters. The information captured will be site and host specific, wherever applicable. The 'types' of information captured, whenever possible, shall include configuration parameters and files; definitions and descriptions of parameters; ECS recommended value; value or value ranges, i.e., common range across multiple DAACs or DAAC-specific range; impacts associated with changing a parameter; and controlling entity (DAAC or Landover) of the parameter.

These baselined documents will be maintained by ECS CM, and are posted on ECS Baseline Information System (EBIS) Site. The baselined information is replicated and posted on a mirror site for DAAC(s) accessibility and review.

All changes to ECS software/hardware patches and versions, are controlled by the EDF and M&O CCBs.

A prototype (non-baselined) version of the configuration information can be posted on EBIS and mirror site for ECS and DAAC review prior to the CCB approval process, as long as, the posted information is clearly identified as 'prototype' to distinguish it from the ECS baselined information.

Depending upon who 'controls' (Landover or DAAC) the configuration parameters, changes to those will follow the appropriate process. The five cases are as follows:

Landover Controlled Parameters: For a 'Landover proposed change' to "Landover controlled" configuration parameters, proposed changes to ECS Baselined Configuration Parameters will proceed through the appropriate EDF CCR process. Changes coming through the ECS Development, SE, CM, Test, Deployment and M&O organizations will pass through the respective CCB(s) for approval. ECS CM will coordinate, document, manage and maintain all the configuration change activities between EDF organization, DAAC(s) and the CCB.

'For a DAAC proposed change' to "Landover controlled" configuration parameter, DAAC(s) will create and submit a CCR. This CCR will be reviewed by the appropriate EDF organizations. EDF engineers will analyze the DAAC proposed change to ECS Configuration Parameters Baseline in terms of a 'DAAC-specific' change, or a 'global change across multiple DAACs'. EDF engineers will assess the need for the proposed change in light of DAAC(s) system operational requirements, subsystem dependencies, DAAC-specific and/or cross-DAAC impacts to the system performance, and overall ECS operational health. EDF will then forward its findings and recommendations to the relevant CCB. The DAAC submitted CCR shall follow the appropriate ECS CCR and CCB approval process (refer to section 3.2). Approved DAAC proposed changes will be updated in the ECS configuration parameter baselined documents, including 910-TDA-022, *Custom Code Configuration Parameters*, which is accessible at <http://cmdm.east.hitc.com/baseline/> under "Technical Documents."

DAAC Controlled Parameters: For 'a Landover proposed change' to "DAAC controlled" configuration parameters, the proposed change will proceed through Landover CCR/CCB process, and will be reviewed and approved by DAAC(s) CCBs/Review Board. Approved Landover proposed changes will be updated in the ECS configuration parameter baselined documents.

For a 'DAAC proposed change' to a "DAAC controlled" parameter, it will proceed through the DAAC change control process and/or DAAC Review Board. The DAAC Review Board assesses the proposed change. If the DAAC Review Board approves the proposed change based on its guidelines, then a notification of the change to the DAAC configuration parameter baseline is sent to the ECS CM. ECSCM will provide the change notice to M&O CCB for informational review. ECS CM will validate such change through its own verification methods

(refer to section 6.8). DAAC approved changes to the DAAC configuration parameters will be updated in the appropriate baselined documents after verification by the ECS CM.

If the DAAC Review Board decides that the 'DAAC proposed change' to a "DAAC controlled" parameter affects the ECS Configuration Baseline, but the proposed change is neither a Trouble Ticket, nor an NCR, then the DAAC will create and submit a CCR. The DAAC CCR will be reviewed by appropriate EDF subsystem engineers in terms of ECS Configuration Parameters Baseline change to that "DAAC-specific" environment, or 'change across multiple DAAC' environments, including possible changes to PVC and VATC. EDF subsystem engineer will forward its recommendation and solution to the relevant CCB. The DAAC submitted CCR shall follow the appropriate ECS CCR and CCB approval process.

Examples of DAAC proposed changes that may affect the ECS Configuration Parameter Baseline are, but not limited to, the following: changes to a configuration parameter setting or a value which is outside of the EDF recommended setting or value range; addition or deletion of a configuration parameter; adding any DAAC unique files/parameters or DAAC unique-extensions which will interface with the ECS custom software, database, OS, COTS and networks; and addition of new hardware, processes, or memory.

Parameter NCR and TT Procedure: Configuration Parameter NCRs can originate both at EDF or DAAC/SMC. NCR originated by an organization within the EDF, or a Trouble Ticket (TT) upgraded to an NCR by the DAAC will be studied by the Problem Review Board, and will follow the NCR process. The responsible engineer or subsystem lead assigned to work the NCR shall identify whether the NCR constitute an ECS Configuration Parameter Baseline change.

Configuration Parameter NCRs originated at EDF will follow the Problem Review Board and EDF CCB approval process.

Configuration Parameter NCR originated from a TT upgrade at the DAAC(s) will follow the Problem Review Board and the M&O CCB approval process

NCRs that affect the ECS configuration parameter baseline, will be entered into DDTS or Remedy(once the transition to Remedy from DDTS is completed) by the responsible engineer, under the 'type' 'configuration parameters'. EDF subsystem engineer responsible for the configuration parameter NCR fix shall document the following "types" of information as part of the solution for NCR: the configuration parameter name; description of the parameter and what is fixed; recommended value or value range; identify if it is a static or a dynamic parameter, if applicable; and possible impacts associated with the configuration of the parameter. In addition, the engineer will provide any implementation guidelines, when necessary.

Final release of the parameter change and updates to the baselined documents will require an approved NCR form with attached redline changes, where applicable. Approved NCR parameter changes will be posted on ECS Baseline Information System (EBIS) and on the DAAC mirror site.

EDF Development Organization Proposed Change Procedure: EDF Development Organization will be responsible for capturing, defining and documenting configuration parameters related to ECS custom code software/hardware, patches/versions, database, operating

systems, COTS and networks. Evolutionary updates to the ECS software/hardware configuration may warrant updates to the ECS Configuration Parameter Baseline documentation. Development will be responsible to provide any such updates using appropriate ECS CCR and CCB approval process. A CCR proposing updates to the ECS Configuration Parameters Baseline documentation shall accompany a red-lined changes, where applicable (red-line changes will not apply when a completely new document is developed), that will clearly identify the following: what baselined document sections are affected, including section(s) that are added, modified or deleted; and additions, modifications or deletions of files and parameters. Development organization shall adhere to the appropriate baselined document 'format' and 'type of information' when providing updates (the 'type' of information is similar to paragraph 6.5.3). ECS CM shall update the relevant baselined documents based on the approved CCR and the attached redlined changes, where applicable, and shall post that updated information on EBIS and DAAC mirror sites.

The EDF Test and Deployment Organizations Proposed Change Procedure: The EDF Test and Deployment Organizations will forward their recommendations to EDF Development in cases when: a certain configuration parameter or setting may be required; a change in value or value range may enhance the performance of the system; or when a parameter setting or value range may pose a risk to the system configuration. Any proposed changes to ECS configuration parameters baseline by the Test/Integration and Deployment Organizations will accompany an analysis report and/or test data (from PVC and VATC) that will be reviewed by the EDF Development engineers. The EDF Test or Deployment shall follow the established ECS CCR and CCB approval process, when required, for proposing a change to the ECS Configuration Parameter Baseline.

ECS CM Organization Responsibilities: The ECS CM responsibility is well defined in document 102-CD-003, Configuration Management Plan for the Science Data Processing Segment of the ECS Project and in other referenced documents. The Configuration Parameter Baseline Maintenance effort falls under the direct scope, responsibility and guidelines described in the above document(s). The ECS CM will maintain 'custom software' configuration parameter baselined information in the EDF Configuration Registry Database. The software and hardware configuration baselined information and its updates will be posted on EBIS and mirror site, as explained in the previous sections of this WI. ECS CM will conduct periodic audits to ensure that the hosts at DAAC(s) and EDF (PVC and VATC) are built to the Configuration Parameter Baseline. In addition, ECS CM will develop 'methods' to capture changes made to the ECS Configuration Parameter Baseline at the DAAC, PVC and VATC without proper notification, or changes that did not follow the appropriate ECS CCR and CCB process. Examples of verification 'methods' are, but not limited to, developing and executing scripts that will take snapshots of DAAC(s), PVC and VTAC modes and hosts at any given time, and running processes to acquire near real-time information and alerts of DAAC(s), PVC and VATC host environment changes.

ECS M&O Responsibilities: The ECS M&O responsibilities are defined in Project Instruction MO-1-003 and other related documents. It will manage the ECS Configuration Parameter Baseline for operations as part of the M&O and/or Sustaining Engineering effort(s). Any proposed change to the ECS Configuration Parameter Baseline for DAAC operations, must be

approved by the M&O CCB. ECS M&O will conduct periodic audits on configuration parameter NCRs in the DDTs. It will review, analyze and validate that approved configuration parameter NCR(s) changes are implemented and documented in the ECS Configuration Parameter Baseline, and it will resolve any discrepancies found. Furthermore, M&O will act as the interface between DAAC and EDF on all issues relating with ECS Configuration Parameters Baseline.

28.6 Overview of Configuration Parameter Files

Various types of source files are used to modify the configuration parameters at the Landover functionality labs and at the DAACs. These are as follows: a) the 'Mkcfg' scripts contain persistent values of configuration parameters, and only ECS developers are allowed to modify them. b) the '.cfgparms', '.extparms' and '.dbparms' files- These files are resident in Clearcase, and contain persistent values of configuration parameters for the EDF. The delivered version of these files to the DAACs contain DAAC specific and mode specific values. The .cfgparms files hold parameters used to generate the standard .CFG files. The .extparms files contain parameters to generate non-standard (or extension)files, and are not used by most subsystems. The .dbparms files hold parameters used in database operations. c) .cfgparms file patches the .cfgparms, .extparms and .dbparms. Changes are from one version to another, e.g., 6A.01 to 6A.02. d) Finally the '.rgypatch' file patches the registry database.

28.6.1 Configuration Registry Documentation

The ECS configuration Registry is intended to improve the manageability of the ECS Custom Code configuration parameters. The concept of the Registry is to provide a centralized service that will be accessible by any application running in the DAAC. Each DAAC will house its own Registry.

The architecture of a Registry will allow hierarchical attribute structure to be represented as well as multi-valued attributes. Detailed information on the Registry Architecture is in CDRL 305, the Registry Database Design is in the CDRL 311-0CD-528 and the Registry GUI description is in CDRL 609.

28.6.2 Overview of Configuration Registry

The Registry Database is initially created with parameters from the .CFG files that are generated by running Ecs Assistant's mkcfg process. In future other files such as the .PCFG and .ACFG files may be housed in the registry database. A .rgypatch file is required to make a change (i.e., to add, update or delete a parameter) to a registry configuration parameter. Another method of making changes to the parameters in the registry is to use the Registry GUI.

ECS servers use a single class to access .CFG files. This class is modified in 5B to request values from the registry servers, if there is no .CFG file present. The registry server then obtains the requested values from the registry database, and returns them to the requester. It is important to note that the registry server provides read-only access to the configuration parameters, and caches non-leaf nodes. Therefore changes made via the GUI which is mode specific, will not take effect until the affected servers have been restarted. If a non-leaf node is added, deleted, or

updated, the changes will not take affect until the registry server, has been restarted, and then the affected servers have been restarted.

The registry schema contains provision for Access Control List (ACL) processing. This capability is scheduled for 6B activity. This would restrict read and write privileges. For example the write privileges of registry in the OPS mode will be granted to individuals at a supervisory role, while read privileges can be non-supervisory etc. Until ACL is implemented, access control to the registry can be managed by database accounts (for

28.6.3 Registry Deployment and Baseline Maintenance

Registry changes can be made by anyone having the DBO privileges. Any changes to the configuration parameter baseline, such as addition, deletion or modifications of parameters, should follow the respective CCB process as mentioned previously in these procedures.

The mode where the registry is to be installed must be running ECS 5B or later. The registry can be brought online either during 5B installation, or shortly after in order to minimize risk.

Run the registry DbBuild script EcCsRgDbBuild to create an empty registry database on the ingest server. On each machine, run the registry population tool (/tools/common/ea/EcCoPopulateRegistry) to copy the configuration parameters from the .CFG files into the registry database. This is a manual process. The tool allows either an individual file or all .CFG files in the directory to be specified.

ECS servers will use the .CFG files, if present in /usr/ecs/<mode>/CUSTOM/cfg when the server is started. Otherwise the ECS servers will use the registry. Therefore move the .CFG files to another directory, and servers will use the registry next time they restart. This allows the registry to be brought online, and taken offline on a server by server basis.

Once the registry has been populated, maintenance of the parameters must be via the registry GUI or via database patches. Changes to the .CFG files or the .cfgparms files will not result in the registry, unless the files are re-imported into the registry. The ability to create a .CFG files during the "make config" phase of Ecs Assist processing can be switched off (via a configuration parameter) after the initial population of the registry. Addition, deletion and modification of values in the registry are achieved either by the GUI or by a database patch script.

The Landover will maintain the "baseline" registry database that will be used to generate the database patches. Landover will no longer provide the .cfgpatch files, and the .cfgparms. The .CFG files will become obsolete.

The baseline registry database will have the structure of a generic DAAC, using functional host names rather than actual host names. It will contain the master values of parameters owned by Landover, and will contain null values for parameters owned by the configuration management process. ECS developers will make changes to the baseline database via the software configuration management process. The database will contain an attribute tree for each release, and release patch supported by Landover. Prior to a release or a release patch, a configuration patch script will be created; by comparing the tree for the new release, with the tree for the release being replaced. The patch script will contain a series of add, update and delete

statements, tagged to indicate the sites they pertain to. The primary purpose of the patch is to enhance the process of parameter additions, modifications or deletions to the DAAC configuration. The patch will also be used to propagate value changes for parameters owned by the Landover.

28.6.4 Registry Preparation Procedures

Create a new Registry Tree for the mode to be installed:

1. Backup the registry database (Use the DAAC general backup procedures, or follow the registry database backup procedures under Section 7.2.).
2. Start the registry GUI for the mode into which the required Drop is being installed.
3. Select the attribute tree name that is mapped to the current mode from the drop-down menu (when the correct tree is selected, the name of the mode should appear in the "Attribute Information" box below).
4. Click on the name of the attribute tree in the window below (the window with a white background), to enable the menu icons.
5. Click on the "Copy selected item" icon (the icon on the left).
6. Enter the name of the new registry tree in the Attribute Name box.
7. Select the <Ok> button.
8. Wait for the new tree to be created (NOTE: It may take a long time for the new tree to be created -- on the order of 30 minutes).
9. Select the newly-created tree name in the window.
10. Click on the MAP icon.
11. Select the mode name from the drop-down menu in the new window that appears and select <Ok>.
12. Exit the GUI by selecting the File → Exit menu option.

28.6.5 Registry Database Backup Procedures

Registry database should be regularly (e.g., nightly) backed up via the Sybase dumps. (The DAACs should have added the registry db to their list of dbs for backup when the registry was installed with 5B.

1. To make a backup copy of a tree within the registry, just click on the "copy" in the registry GUI before making modifications to the tree.
2. To make a backup copy of a tree and dump it to a file, click on "Add New Tree" at the bottom of the tree display in the GUI. Then choose the MkRgyPatch option from the menu,

and specify the new tree (which is empty) and the tree that you want to copy. This will produce a file containing the tree in the rgypatch format.

Caution: Check before deleting a tree, as this may hose the registry db. A fix to this bug will be provided to the DAACs shortly.

28.6.6 Registry Patch Procedures

Patch the Registry Database using the **.rgypatch** file:

1. From the ECS Assist Subsystem Manager, select the appropriate Mode, Subsystem, and Component from the main window.
2. Select "Registry Data Patch" from the "Tools" menu. An "Apply Registry Data Patch" window will appear.
3. In the "Apply Registry Data Patch" window, enter the name of the SQL server in the "Registry Database Server:" box.
4. Enter the registry database DBO ID and password, respectively, in the next two boxes.
5. In the next ("Registry DB Name:") box, be sure to enter the name of the registry database (**EcCsRegistry[_<mode>]**) and press the <ENTER> key. This will cause ECS Assist to connect to the registry database and populate the drop-down menu associated with the next field ("Tree to patch:").
6. Use the drop-down menu to select the appropriate registry tree that is being patched (if you are not sure which tree to select, bring up the registry GUI and verify which tree is mapped to the mode that you are updating).
7. Finally, click on the "Select Patch File" button to bring up the "File Selection Dialog" window.
8. Navigate through this window to find the .rgypatch file (note: if your installation was successful, it should appear in the /usr/ecs/<MODE>/CUSTOM/.installed/DMS directory).
9. Highlight the .rgypatch file in the window and select <OK>.
10. Verify that the appropriate information is indicated in the "Patch File:" box in the "Apply Registry Data Patch" window and select <OK>. At this point, the registry patch will be applied.
11. Monitor the output via ECS Assist for any warning or error messages as the patch is run.

This page intentionally left blank.

Appendix A. Additional Material

Examples of the Various ODL Files Used by Each Instrument Team

Section 26.13 deals, in part, with the use of ODL files in SSI&T activities. This Appendix serves as a supplement and reference for that section. Useful examples of ODL files follow. ODL Template files, from which specific examples were created, are listed first. Then, examples of specific ODL files are listed by instrument (ASTER, MISR, MODIS and AIRS). *Please note that in many of the examples that follow, much of the instrument/ECS provided comments have been deleted in order to keep this document reasonably short.*

A.1 Template ODL Files

There are five Template ODL files listed here. The specific or tailored ODL files listed in Sections A.2 through A.5 were derived from these templates by appropriate editing and filling-in of values (**NOTE: while the TILE ODL file is currently not being used by any of the instrument teams mentioned above, the template is included here for completeness*). The five ODL Template files listed reside, on the AIT Sun host, at /usr/ecs/<mode>/CUSTOM/data/DPS. They are:

PGE_ODL.template
ESDT_ODL.template
ORBIT_ODL.template
TILE_ODL.template*
PATHMAP_ODL.template

A.1.1 PGE_ODL.template

```
/* **** */
/*
/*          TEMPLATE PGE SCIENCE METADATA ODL FILE          */
/*
/*          */
/*          */
/* The SSIT operator's responsibility is to copy this file over and */
/* edit it to add all necessary PDPS metadata values.          */
/*          */
/* All PGE ODL files must reside in directory $DPAT_PGE_SCIENCE_MD. */
/* This directory is now set through the Process Framework CFG files. */
/*          */
/* The operator must add a value to the right of the "=" for each */
/* parameter.          */
/*          */
/* Normally, a template version of this file (without the comments) */
/* will be generated by the SSIT operator from the PCF delivered to */
/* SSIT. This file is meant to show the SSIT personnel and the */
/* Instrument teams the information that is needed for a PGE to be */
/* planned and executed by the Planning and Data Processing system of */
/* ECS.          */
```

```

/*
/* CHANGE LOG
/*
/* -- Added new schedule type for Data Scheduled PGEs. 02/18/98 */
/* Changed QUERY_DELAY to be optional for all PGEs. */
/* Changed SPATIAL_KEY_INPUT to KEY_INPUT */
/* -- Fixed description for Begin/End Period Offsets. 03/10/98 */
/* -- Added The Distinct Value definition. 03/26/98 */
/* -- Fixed length of CATEGORY. 03/27/98 */
/* -- Fixed length of FILETYPE_NAME. 03/31/98 */
/* -- Added PATHMAP_NAME 04/13/98 */
/* -- Changed how WAITFOR is supposed to be set. 05/06/98 */
/* Added entries for ASSOCIATED_SCIENCE_DATA to handle */
/* BROWSE and QA products. */
/* -- Added START_OF_MINUTE to PROCESSING_BOUNDARY. 06/24/98 */
/* Updated DATA DAY values for PGE_PARAMETER_DYNAMIC_VALUE, */
/* and DATABASE_QUERY */
/* -- Added KEY_PARAMETER_NAME and KEY_PARAMETER_VALUE 07/05/98 */
/* for Metadata Checks and Metadata Queries. */
/* -- Updated description for KEY_PARAMETER_NAME and 07/11/98 */
/* KEY_PARAMETER_VALUE. */
/* -- Updated lengths for PLATFORM and INSTRUMENT. 08/13/98 */
/* -- Updated explanation for "Already Created Tile" 08/18/98 */
/* for QUERY_TYPE. */
/* -- Added CHECK_FOR_OUTPUT flag. 08/24/98 */
/* -- Added MOST_RECENT_QUERY_OFFSET and MOST_RECENT_ 09/02/98 */
/* QUERY_RETRIES parameters for the Most Recent Granule */
/* Production Rule. */
/* -- Added AUXILIARY_LOGICAL_ID object for handling_ 09/23/98 */
/* multiple L0 granules. */
/* Removed older change commentary. */
/* -- Added COMPOUND_PGE parameter for handling 10/23/98 */
/* PGEs with multiple executables. Also deleted */
/* old change history. */
/* -- Updated description for ALTERNATE_INPUT_TIMER 11/07/98 */
/* to say that it has not affect for Dynamic */
/* Internal ESDTs. */
/* -- Added ALIGN_DPR_TIME_WITH_INPUT_TIME parameter 12/20/98 */
/* -- Increased the number of Profile Ids from 99 to 07/12/99 */
/* 999. */
/* Removed restriction on ALTERNATE_INPUT_TIMER with respect */
/* to Internal Dynamic ESDTs. */
/* Added PGE_DEFAULT_PROFILE parameter. */
/* Added PROFILE_SELECTOR_PGE_PARAMETER. */
/* -- Added Closest Granule values: 08/19/99 */
/* CLOSEST_QUERY_OFFSET, CLOSEST_QUERY_RETRIES, and */
/* CLOSEST_QUERY_DIRECTION. */
/* -- Added "Metadata" to the query type. 12/13/99 */
/* -- Updated Toolkit logical Ids that SSIT allows 02/03/00 */
/* in ODL. */
/* Removed old change history */
/*
/*****

/*****
/* PGE name */
/* -- Must be a string, max len 10 characters */
/* -- PGE name inside ODL file must be identical to */

```



```

/*          PGE name used as part of ODL filename          */
/*          Example                                          */
/*          PGE_NAME = "ssit"                              */
/*****

PGE_NAME = ""

/*****
/*          PGE version                                    */
/*          -- Must be a string, max len 5 characters      */
/*          -- PGE version inside ODL file must be identical to */
/*          PGE version used as part of ODL filename      */
/*          Example                                          */
/*          PGE_VERSION = "1.0"                            */
/*****

PGE_VERSION = ""

/*****
/*          PGE Profile ID                                */
/*          -- Must be an integer                          */
/*          -- Must be >= 0 and <= 999                    */
/*          Example                                          */
/*          PROFILE_ID = 99                                */
/*****

PROFILE_ID =

/*****
/*          PGE Profile Description                        */
/*          -- Must be a string, max length 255 characters */
/*          Example                                          */
/*          PROFILE_DESCRIPTION = "Improved performance numbers" */
/*****

PROFILE_DESCRIPTION = ""

/*****
/*          PGE On-Demand Profile Default                 */
/*          -- Must be a string, set to "Y" or "N".        */
/*          -- If NOT Present, defaults to "N".            */
/*          -- Marks a particular for this PGE (PGE Name + */
/*          PGE version) as the default for On Demand Processing */
/*          Requests.                                       */
/*          -- If more than 1 PGE (PGE Name + PGE Version) has this */
/*          value set, an error will be returned.         */
/*          Example                                          */
/*          PGE_DEFAULT_PROFILE = "N"                      */
/*****

PGE_DEFAULT_PROFILE = ""

/*****
/*          Spacecraft platform name                      */
/*          -- Must be a string, max len 25 characters     */
/*          Example                                          */
/*          PLATFORM = "TRMM"                              */
/*****

```

```

PLATFORM = ""

/*****
/*      Instrument name                                */
/*      -- Must be a string, max len 20 characters    */
/*      Example                                         */
/*      INSTRUMENT = "CERES"                           */
*****/

INSTRUMENT = ""

/*****
/*      Minimum Number of Outputs                      */
/*      (used for QA purposes)                         */
/*      -- Must be a integer, maxium 3 digits.        */
/*      Example                                         */
/*      MINIMUM_OUTPUTS = 0                            */
*****/

MINIMUM_OUTPUTS =

/*****
/*      Type of PGE Scheduling                          */
/*      -- Must be a string with one of the following values: */
/*      "Time" = TimeScheduled (PGE is scheduled based on the */
/*      boundary/period and the arrival of data).          */
/*      "Data" = DataScheduled (PGE is scheduled based on the */
/*      avialability of data produced by other             */
/*      PGEs).                                             */
/*      "Tile" = TileScheduled (PGE is scheduled based on the */
/*      the definition of Tiles). Note that                */
/*      TILE_SCHEME_NAME must have a value for Tile       */
/*      Scheduled PGEs.                                    */
/*      "Orbit" = OrbitScheduled (PGE is scheduled based    */
/*      the orbit of the spacecraft. Note that then       */
/*      PROCESSING_PERIOD must = "ORBITS=1" and           */
/*      PROCESSING_BOUNDARY must =                        */
/*      "START_OF_ORBIT". Also, A file of named           */
/*      ORBIT_<platform>.odl must be present.             */
/*      Also if you want a Pathmap it needs to be         */
/*      specified under PATHMAP_NAME.                     */
/*      "Snapshot" = SnapshotScheduled (PGE is scheduled   */
/*      based on a single date/time entered                */
/*      entered when the production request is            */
/*      submitted.                                         */
/*      Example                                           */
/*      SCHEDULE_TYPE = "Tile"                            */
*****/

SCHEDULE_TYPE = ""

/*****
/*      Nominal time interval between start of PGE runs */
/*      -- NOT needed for PGEs where SCHEDULE_TYPE = "Snapshot" */
/*      or SCHEDULE_TYPE = "Data".                        */
/*      -- Must contain a single P=V string, where       */
/*      P is one of { YEARS, MONTHS, THIRDS WEEKS, DAYS, */

```

```

/*          HOURS, MINS, SECS, ORBITS} */
/*          -- NOTE that ORBITS must be used for PGEs based on an */
/*          Orbit Model. Note that PROCESSING_BOUNDARY must be */
/*          set to "START_OF_ORBIT". */
/*          Example */
/*          PROCESSING_PERIOD = "DAYS=1" */
/*****/

PROCESSING_PERIOD = ""

/*****/
/*          Nominal time boundary on which PGE processing begins */
/*          -- NOT needed for PGEs where SCHEDULE_TYPE = "Snapshot" */
/*          or SCHEDULE_TYPE = "Data". */
/*          -- Must contain a one of */
/*          { START_OF_MINUTE, START_OF_HOUR, START_OF_6HOUR, */
/*            START_OF_DAY, START_OF_WEEK, */
/*            START_OF_ONE_THIRD_MONTH, */
/*            START_OF_MONTH, START_OF_YEAR, START_DATE, */
/*            START_OF_ORBIT }; */
/*          also, "+<n>" or "-<n>" may be added to any of these, */
/*          where <n> specifies integer seconds. */
/*          For START_DATE an "=" can be added followed by the */
/*          start date. */
/*          -- NOTE that START_OF_ORBIT must be used for PGEs based */
/*          on an Orbit Model. A file of named */
/*          ORBIT_<platform>.odl must be present. */
/*          Example */
/*          PROCESSING_BOUNDARY = "START_OF_HOUR" */
/*****/

PROCESSING_BOUNDARY = ""

/*****/
/*          Software version */
/*          -- Must be a string, max 5 len characters */
/*          -- If Ssw version is not the same as PGE version, */
/*          SswId ("<PGE Name>#<Ssw Version>") must already */
/*          be defined in the database; */
/*          That is, the only allowed values of the */
/*          software version are either this PGE version */
/*          or a previous PGE version for this PGE name */
/*          Example */
/*          PGE_SSW_VERSION = "1.0" */
/*****/

PGE_SSW_VERSION = ""

/*****/
/*          Delay for query */
/*          -- Optional for types of PGEs. */
/*          -- The amount of time (in SECONDS) that the query for */
/*          input data should be delayed. This value is added */
/*          onto the Stop Time of any DPR generated with this */
/*          PGE. */
/*          -- Used for Tiling or Metadata Query inputs. */
/*          -- OPTIONAL Parameter. If not specified it is set to 0. */
/*          -- Must be an integer value >= 0. */

```

```

/*          Example                                     */
/*          QUERY_DELAY = 360 (1 hour)                 */
/*                                                     */
/*****

QUERY_DELAY = 0

/*****
/*          Name of the Tiling Scheme used             */
/*          -- Must be a string of at most 20 characters. */
/*          -- There can be NO spaces in the string.      */
/*          -- A file that defines the Tiling Scheme must */
/*          be created with the name TILE_<tiling scheme>.odl */
/*          Example                                     */
/*          TILE_SCHEME_NAME = "Earth_Squared"          */
/*                                                     */
/* NOTE that this is only needed for PGEs of Schedule Type = "Tile". */
/* It can be deleted for all other types of PGEs.          */
/*****

TILE_SCHEME_NAME = ""

/*****
/*          Name of Pathmap used                       */
/*          -- Must be a string of at most 25 characters. */
/*          -- There can be NO spaces in the string.      */
/*          -- A file that defines the Pathmap must       */
/*          be created with the name PATHMAP_<Pathmap_Name>.odl */
/*          Example                                     */
/*          PATHMAP_NAME = "Some_Name"                  */
/*                                                     */
/* NOTE that this is only needed for PGEs of Schedule Type = "Orbit". */
/* It can be deleted for all other types of PGEs.          */
/*****

PATHMAP_NAME = ""

/*****
/* OPTIONAL PARAMETER                                 */
/*          Check For Outputs                          */
/*          -- Must be a character value of either "Y" (YES) */
/*          or "N" (NO).                                     */
/*          -- Defaults to "N" if not specified.            */
/*          -- When set to "Y", this means that a DPR of the PGE */
/*          will ONLY be scheduled if the output of that PGE has */
/*          NOT been produced. This is currently planned for use */
/*          in ASTER Routine Processing.                    */
/*          -- Note that creating a DPR (in the Production Request */
/*          Editor) with Reprocessing set will override this */
/*          flag.                                           */
/*          Example                                     */
/*          CHECK_FOR_OUTPUTS = "N"                      */
/*****

CHECK_FOR_OUTPUTS = "N"

/*****
/* OPTIONAL PARAMETER                                 */

```

```

/*      Compound Pge Flag                                */
/*      -- Must be a character value of either "Y" (YES) */
/*      or "N" (NO).                                    */
/*      -- Defaults to "N" (Not Compound PGE) if not specified. */
/*      -- When set to "Y", this means that this PGE is made up */
/*      of multiple executables AND that the output of one      */
/*      of these executables is the input of another            */
/*      executable within the PGE.                              */
/*      -- Note that setting this flag will hurt the performance */
/*      of the Destaging step during PGE execution. It is      */
/*      best to only set it to "Y" if both conditions          */
/*      mentioned above are true.                                */
/*      Example                                                  */
/*      COMPOUND_PGE = "N"                                       */
/*****

COMPOUND_PGE = "N"

/*****
/* Exit message object */
/* */
/* Defines a possible PGE exit code, and associates a message with it. */
/* */
/* This object is optional and can be deleted if no EXIT MESSAGEs are */
/* desired. */
/* */
/* Replicate the object as needed to define EXIT MESSAGEs for multiple */
/* EXIT CODEs. */
/* */
/* See "Establishing Science Software Exit Conditions for the */
/* Production Environment" white paper (420-WP-006-002) for the */
/* definitions and of exit code values and their uses. */
/* */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****

OBJECT = EXIT_MESSAGE

/*****
/*      Class (object counter, used only to distinguish objects) */
/*      -- Must be an integer */
/*      -- Must be unique in this file for this type of object */
/*      -- Must be greater than 0. */
/*      Example */
/*      CLASS = 1 */
/*****

CLASS= 1

/*****
/*      Exit code for this PGE */
/*      -- Must be an integer */
/*      -- Must be 0 or between 200 and 239 */
/*      Example */
/*      EXIT_CODE = 200 */
/*****

EXIT_CODE = 0

```

```

/*****
/*      Message corresponding to this exit code      */
/*      -- Must be a string, max len 240 characters  */
/*      Example                                     */
/*      EXIT_MESSAGE = "PGE successfully completed" */
*****/

EXIT_MESSAGE = ""

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
*****/

END_OBJECT = EXIT_MESSAGE

/*****
/* Exit dependency object                                     */
/*                                                         */
/* Defines names, exit codes and conditions of PGEs on which this */
/* PGE depends.                                             */
/*                                                         */
/* This object is optional and can be deleted if no EXIT DEPENDANCY(s) */
/* exist for this PGE.                                     */
/*                                                         */
/* Replicate this object as needed to define multiple EXIT */
/* DEPENDANCies for the PGE.                               */
/*                                                         */
/* See "Establishing Science Software Exit Conditions for the */
/* Production Environment" white paper (420-WP-006-002) for the */
/* definitions and of exit code values and their uses.      */
/*                                                         */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
*****/

OBJECT = EXIT_DEPENDENCY

/*****
/*      Class (object counter, used only to distinguish objects) */
/*      -- Must be an integer                                     */
/*      -- Must be unique in this file for this type of object */
/*      -- Must be greater than 0.                               */
/*      Example                                                 */
/*      CLASS = 1                                              */
*****/

CLASS= 1

/*****
/*      Name of PGE upon which this PGE depends      */
/*      -- Must be a string, max len 10 characters    */
/*      -- SswId ("<PGE name>#<Ssw version>") must be different */
/*      than this SswID (PGE cannot depend on itself) */
/*      -- SswId must already exist in the database    */
/*      Example: This CERES PGE depends on the exit code of */
/*      a MODIS PGE: execute the CERES PGE only if the */
/*      MODIS PGE had exit code = 0                    */
/*      DEPENDENCY_PGE_NAME = "MODIS"                  */
*****/

```

```

/*****/

DEPENDENCY_PGE_NAME = ""

/*****/
/*      Version of Ssw upon which this Ssw depends      */
/*      -- Must be a string, max len 5 characters      */
/*      -- SswId ("<PGE name>#<Ssw version>") must be different */
/*      than this SswID (PGE cannot depend on itself) */
/*      Example                                          */
/*      DEPENDENCY_SSW_VERSION = "x"                    */
/*****/

DEPENDENCY_SSW_VERSION = ""

/*****/
/*      Operator for exit code dependency condition      */
/*      -- Must be one of { >, <, >=, <=, =, != }      */
/*      Example                                          */
/*      EXIT_OPERATION = "="                            */
/*****/

EXIT_OPERATION = ""

/*****/
/*      Exit code for PGE upon which this PGE depends  */
/*      -- Must be an integer                          */
/*      -- Must be 0 or between 200 and 239            */
/*      -- Must already exist in the database as a valid */
/*      exit code for the PGE upon which this PGE depends */
/*      Example                                          */
/*      EXIT_CODE = 0                                   */
/*****/

EXIT_CODE = 0

/*****/
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****/

END_OBJECT = EXIT_DEPENDENCY

/*****/
/* PCF entry object                                          */
/*                                                         */
/* The program DpAtCreateOdlTemplate (run at SSIT) generates one of */
/* these object for each file entry in the PCF. Only generic Toolkit */
/* Logical IDs are ignored during Template Creation.              */
/*                                                         */
/* The operator needs to fill in values for the parameters as described */
/* in the comments for each parameter. Note that some parameters */
/* must be filled for each PCF entry, while others are optional or only */
/* needed based on the values of other parameters.              */
/*                                                         */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED                    */
/*****/

OBJECT = PCF_ENTRY

```

```

/*****
/*      Class (object counter, used only to distinguish objects) */
/*      -- This line is generated by DpAtCreateOdlTemplate */
/*      from the PCF and is normally not modified */
/*      -- Must be an integer */
/*      -- Must be unique in this file for this type of object */
/*      -- Must be greater than 0. */
/*      Example */
/*      CLASS = 1 */
*****/

CLASS = 1

/*****
/*      PCF logical ID */
/*      -- This line is generated by DpAtCreateOdlTemplate */
/*      from the PCF and is normally not modified. */
/*      -- Must be a positive integer. */
/*      -- Most values between 10000 and 10999 (Toolkit specific */
/*      Logical IDs) are ignored except for the following: */
/*      Data Dictionary Logical ID (10251) */
/*      Attitude Data Logical ID (10501) */
/*      Ephmerous Data Logical ID (10502) */
/*      Math Constant Logical ID (10999) */
/*      Index Data File Logical ID (10900) */
/*      DEM Logical Ids (10649 - 10655) */
/*      Ascii Dump Logical ID (10255) */
/*      Disable Status Level RTI Logical ID (10117) */
/*      Disable Seed RTI Logical ID (10118) */
/*      Disable Status code RIT Logical ID (10119) */
/*      Example */
/*      LOGICAL_ID = 100 */
*****/

LOGICAL_ID = 100

/*****
/*      PCF file type */
/*      -- This line is generated by DpAtCreateOdlTemplate */
/*      from the PCF and is normally not modified */
/*      -- Must be an integer between 1 and 8 inclusive */
/*      =1, PRODUCT INPUT FILES */
/*      =2, PRODUCT OUTPUT FILES */
/*      =3, SUPPORT INPUT FILES */
/*      =4, SUPPORT OUTPUT FILES */
/*      =5, USER DEFINED RUNTIME PARAMETERS */
/*      =6, INTERIM/INTERMEDIATE INPUT FILES */
/*      =7, INTERIM/INTERMEDIATE OUTPUT FILES */
/*      =8, TEMPORARY I/O */
/*      Example */
/*      PCF_FILE_TYPE = 1 */
*****/

PCF_FILE_TYPE = 1

/*****
/*      Data Type Name -- same as Data Server ESDT Short Name */
*****/

```



```

/*      -- Must be a string, max len 8 characters */
/*      -- Required for all PCF ENTRY objects, except those with */
/*      PCF_FILE_TYPE = 5 or 8 */
/*      -- An ESDT_ODL file for this name must exist in */
/*      in directory $DPAT_ESDT_SCIENCE_MD, and have a name */
/*      of the form */
/*      "ESDT_<Data Type Name#Data Type Version>.odl" */
/*      -- An ESDT of this Short Name must already be defined */
/*      at the Data Server */
/*      Example */
/*      DATA_TYPE_NAME = "TRWpl182" */
/*      implies file $DPAT_ESDT_SCIENCE_MD/ESDT_TRWpcal182.odl */
/*      already exists in the SSIT environment, and that */
/*      ESDT Short Name "TRWpl182" already exists in the */
/*      Data Server */
/*      */
/*****

DATA_TYPE_NAME = ""

/*****
/*      Data Type Version */
/*      -- Must be a string, max len 5 characters */
/*      -- Required for all PCF ENTRY objects, except those with */
/*      PCF_FILE_TYPE = 5 or 8 */
/*      -- An ESDT_ODL file for this name must exist in */
/*      in directory $DPAT_ESDT_SCIENCE_MD, and have a name */
/*      of the form */
/*      "ESDT_<Data Type Name#Data Type Version>.odl" */
/*      -- An ESDT of this Short Name and Version must already */
/*      be defined at the Data Server */
/*      Example */
/*      DATA_TYPE_VERSION = "3.5.1" */
/*****

DATA_TYPE_VERSION = ""

/*****
/*      Minimum number of input granules for this logical ID */
/*      -- This line is generated by DpAtCreateOdlTemplate */
/*      from the PCF and is only modified if the PGE can */
/*      execute successfully with fewer granules than in the */
/*      PCF from which the template was generated. */
/*      -- Used to support "Minimum Number of Granules" */
/*      Production Rule. */
/*      -- Required for all PCF ENTRY objects */
/*      PCF_FILE_TYPE = 1, 3, 6 (ignored otherwise). */
/*      -- Must be a >= 0. */
/*      -- Note that for number of files within a granule */
/*      greater than one, the FILE TYPE object for this entry */
/*      must be changed to specify the various file types and */
/*      maximum number of files. */
/*      Example */
/*      MIN_GRANULES_REQUIRED = 1 */
/*****

MIN_GRANULES_REQUIRED = 1

/*****

```

```

/*      Maximum number of input granules for this logical ID      */
/*      -- This line is generated by DpAtCreateOdlTemplate */
/*      from the PCF and is only modified if the PGE can      */
/*      execute successfully with more granules than in the */
/*      PCF from which the template was generated.      */
/*      -- Used to support "Minimum Number of Granules"      */
/*      Production Rule.      */
/*      -- Required for all PCF ENTRY objects      */
/*      PCF_FILE_TYPE = 1, 3, 6 (ignored otherwise)      */
/*      -- Must be a positive integer      */
/*      -- Note that for number of files within a granule */
/*      greater than one, the FILE TYPE object for this entry */
/*      must be changed to specify the various file types and */
/*      maximum number of files.      */
/*      Example      */
/*      MAX_GRANULES_REQUIRED = 1      */
/*****/

MAX_GRANULES_REQUIRED = 1

/*****/
/*      Begin Period Offset.      */
/*      -- Only needed if data for this PCF entry is to be */
/*      selected BEFORE (-) or AFTER (+) the period defined */
/*      for the ESDT (stated in the corresponding ESDT      */
/*      ODL file).      */
/*      -- Defaulted to 0.      */
/*      -- If set, must be an integer number of seconds.      */
/*      A positive value indicates that the value is BEFORE */
/*      the Period of the ESDT. A Negative value is added to */
/*      the Period so that the data will be found after the */
/*      start of the period specified for the ESDT.      */
/*      Example      */
/*      BEGIN_PERIOD_OFFSET = "7200" (2 hours)      */
/*****/

BEGIN_PERIOD_OFFSET = 0

/*****/
/*      End Period Offset.      */
/*      -- Only needed if data for this PCF entry is to be */
/*      selected BEFORE (-) or AFTER (+) the period defined */
/*      for the ESDT (stated in the corresponding ESDT      */
/*      ODL file).      */
/*      -- Defaulted to 0.      */
/*      -- If set, must be an integer number of seconds.      */
/*      A positive value indicates that the value is AFTER */
/*      the Period of the ESDT. A Negative value is */
/*      subtracted from the end of the period to find data */
/*      starting within the period specified for the ESDT. */
/*      Example      */
/*      END_PERIOD_OFFSET = "-7200" (2 hours)      */
/*****/

END_PERIOD_OFFSET = 0

/*****/
/*      Input file group ID      */

```

```

/*          -- Required for all PCF ENTRY objects with          */
/*          PCF_FILE_TYPE = 1, 3, 6 (ignored otherwise).        */
/*          -- Only used when input is defined as Static in ESDT */
/*          ODL.                                                */
/*          -- Must be a string                                */
/*          -- 1st character must be one of {C,L,D,O}          */
/*          C -- Coefficient file                                */
/*          L -- Lookup file                                    */
/*          D -- Database file                                  */
/*          O -- Other Type file                                */
/*          -- Rest of string must resolve to a                */
/*          positive integer < 10000                            */
/*          Example                                             */
/*          SCIENCE_GROUP = "C1"                                */
/*****

SCIENCE_GROUP = ""

/*****
/*          Type of Input                                     */
/*          -- Required for all PCF ENTRY objects with          */
/*          PCF_FILE_TYPE = 1,3,6 (ignored otherwise)          */
/*          -- Must be a string with one of the following values: */
/*          "Required" = Required input/no alternates          */
/*          "Primary" = Primary input/alternates defined        */
/*                      Alternate_Input object defined for this */
/*                      PCF Entry.                                */
/*          "Optional" = Optional input, PGE can run without it. */
/*                      An Optional_Input object must be defined */
/*                      for this PCF Entry.                      */
/*          "Alternate" = Alternate input/there will be an      */
/*                      Alternate_Input object defined for this */
/*                      PCF Entry.                                */
/*          Example                                             */
/*          INPUT_TYPE = "Required"                             */
/*****

INPUT_TYPE = ""

/*****
/*          Align DPR Time with Input                         */
/*          -- Specifies that the time of the DPR will be shifted */
/*          to match the real time of input for this Logical Id. */
/*          -- May only be set for one input per PGE Profile.    */
/*          -- Valid values are "Y" or "N".                      */
/*          -- If not specified, it is set to "N".              */
/*          Example                                             */
/*          ALIGN_DPR_TIME_WITH_INPUT_TIME = "Y"               */
/*****

ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"

/*****
/*          Number of Alternate Inputs needed.                 */
/*          -- Required for all PCF ENTRY objects with          */
/*          PCF_FILE_TYPE = 1,3,6 that have                    */
/*          INPUT_TYPE = "Primary" (ignored otherwise)          */
/*          -- Must be either 0 or 1.                            */

```

```

/*          Example                                     */
/*          NUMBER_NEEDED = 1                           */
/*          (This means that only 1 of the alternate inputs is */
/*          required to execute the PGE)                  */
/*****

NUMBER_NEEDED =

/*****
/*          Distinct Value for the input.                  */
/*          -- Optional entry for PCF ENTRY objects with */
/*          PCF_FILE_TYPE = 1,3,6. Set to null if not provided. */
/*          -- A string value, max length 80 characters. */
/*          -- A value that will allow unique naming of granules */
/*          input by a PGE. */
/*          -- Must be the name of a metadata parameter defined in */
/*          a METADATA_DEFINITION object. If a parameter is */
/*          is specified for which no METADATA_DEFINITION object */
/*          exists an error will be raised during ODL parsing. */
/*          -- Supports what are called Multi-Granule ESDTs. These */
/*          are ESDTs that have multiple granules for the same */
/*          time period where the only difference between the */
/*          granules is metadata parameters. */
/*          Example */
/*          DISTINCT_VALUE = "CAMERA_DF" */
/*****

DISTINCT_VALUE = ""

/*****
/*          Query Type for the input.                  */
/*          -- Optional entry for PCF ENTRY objects with */
/*          PCF_FILE_TYPE = 1,3,6. */
/*          -- Must be one of */
/*          "Temporal" -- Data is retrieved by time. */
/*          "Spatial" -- Data is retrieved by spatial location */
/*          of 'key' data type. */
/*          "Tile" -- Data is retrieved by spatial location */
/*          of the tile. */
/*          "Already Created Tile" */
/*          -- Data is retrieved by query of tiles */
/*          already produced (used for cases when */
/*          one PGE needs the tile output of one or */
/*          more other PGEs). */
/*          "Metadata" -- Data is retrieved via temporal query and */
/*          a metadata query */
/*          -- NOTE that if "Already Created Tile" is used, then */
/*          a Metadata Query is expected to query on the TileId */
/*          parameter in the metadata. "Already Created Tile" */
/*          will NOT work without a metadata parameter that holds */
/*          the TileId. */
/*          -- The default is "Temporal" (if not specified). */
/*          Example */
/*          QUERY_TYPE = "Temporal" */
/*****

QUERY_TYPE = ""

```

```

/*****
/*      Spatial Time Delta.                                     */
/*      -- Required for PCF ENTRY objects with                  */
/*      PCF_FILE_TYPE = 1,3,6 that have QUERY_TYPE =           */
/*      "Spatial".                                              */
/*      -- An Integer that allows for some time differential    */
/*      when querying for input data on spatial constraints.    */
/*      It is added to the Start/Stop times of the DPR.        */
/*      -- Time is specified in seconds                          */
/*      Example                                                 */
/*      SPATIAL_TIME_DELTA = 100                                */
*****/

```

SPATIAL_TIME_DELTA =

```

/*****
/*      Spatial Pad                                             */
/*      -- Required for PCF ENTRY objects with                  */
/*      PCF_FILE_TYPE = 1,3,6 that have QUERY_TYPE =           */
/*      "Temporal".                                             */
/*      -- A real number (float) value equal to 0.0 or 1000.0.  */
/*      Or, a value between those endpoints. The units of      */
/*      measure is kilometers. INTEGERS are not valid!         */
/*      (i.e. 10, 500)                                          */
/*      -- This pad will be applied to the KEY INPUT granule   */
/*      Example                                                 */
/*      SPATIAL_PAD = 100.0                                     */
*****/

```

SPATIAL_PAD =

```

/*****
/*      Key Input Data Type.                                     */
/*      -- Optional for PCF ENTRY objects with                  */
/*      PCF_FILE_TYPE = 1,3,6 that have QUERY_TYPE =           */
/*      "Temporal" (ignored otherwise).                          */
/*      -- Specifies one of the following:                       */
/*      -- Spatial constraints of this input should be         */
/*      used when acquiring all data with QUERY_TYPE =         */
/*      "Spatial".                                              */
/*      -- The number of granules for the input should         */
/*      determine if a DataScheduled PGE should be             */
/*      run.                                                    */
/*      -- Must be one of "Y" or "N".                           */
/*      -- "YES" should only be set for a single input with a  */
/*      QUERY_TYPE = "Temporal".                                */
/*      -- NOTE that the old version of this parameter         */
/*      SPATIAL_KEY_INPUT is still supported and will be       */
/*      treated as having the same meaning.                     */
/*      Example                                                 */
/*      KEY_INPUT = "Y"                                         */
*****/

```

KEY_INPUT = ""

```

/*****
/*      OPTIONAL PARAMETER                                     */
/*      Query Offset for Closest Granule.                       */
*****/

```

```

/*      -- Optional entry for PCF ENTRY objects with      */
/*      PCF_FILE_TYPE = 1,3,6. Set to 0 if not provided.    */
/*      -- Must contain a single P=V string, where          */
/*      P is one of {WEEKS, DAYS, HOURS, MINS, SECS}.       */
/*      Other valid period values are NOT supported for this */
/*      parameter.                                           */
/*      -- Used if input is expected to be the "Closest Granule". */
/*      This means that the data under this PCF ENTRY will be */
/*      queried for every CLOSEST_QUERY_OFFSET from the      */
/*      Start Time of the Data Processing Request for the PGE, */
/*      either forward or backward as indicated by the value */
/*      of CLOSEST_QUERY_DIRECTION.                         */
/*      -- Closest Granule supercedes Most Recent Granule   */
/*      Example                                              */
/*      CLOSEST_QUERY_OFFSET = "DAYS=1"                    */
/*****

CLOSEST_QUERY_OFFSET =

/*****
/*      Closest Granule Direction.                        */
/*      -- Required for PCF ENTRY objects with            */
/*      PCF_FILE_TYPE = 1,3,6 that have specified         */
/*      CLOSEST_QUERY_OFFSET.                             */
/*      -- A string that indicates the direction of a search */
/*      for a desired granule. Must be either:             */
/*      "Forward" or "Backward"                           */
/*      -- CLOSEST_QUERY_DIRECTION determines the direction */
/*      of search (timewise) to query for a suitable granule */
/*      from the Start Time of the Data Processing Request */
/*      for the PGE, either forward or backward.          */
/*      -- Closest Granule supercedes Most Recent Granule   */
/*      Examples                                           */
/*      CLOSEST_QUERY_DIRECTION = "Forward"                */
/*      CLOSEST_QUERY_DIRECTION = "Backward"               */
/*****

CLOSEST_QUERY_DIRECTION =

/*****
/*      Closest Granule Maximum Number of Retries.        */
/*      -- Required for PCF ENTRY objects with            */
/*      PCF_FILE_TYPE = 1,3,6 that have specified         */
/*      CLOSEST_QUERY_OFFSET.                             */
/*      -- An Integer that allows a number of retries on the */
/*      inputs where the "Closest Granule" is expected.     */
/*      -- The Query Offset set in the above parameter      */
/*      (CLOSEST_QUERY_OFFSET) is used to repeat the       */
/*      the query for the data for for time periods of     */
/*      Query Offset starting from the Start Time of the   */
/*      Data Processing Request for the PGE either forward or */
/*      backward as indicated by the value                 */
/*      of CLOSEST_QUERY_DIRECTION.                       */
/*      -- Closest Granule supercedes Most Recent Granule   */
/*      Example                                              */
/*      CLOSEST_QUERY_RETRIES = 20                         */
/*****

```

```

CLOSEST_QUERY_RETRIES =

/*****
/* File Types Object */
/*
/* THIS OBJECT IS REQUIRED for PCF_FILE_TYPES = 1, 2, 3, 4, 5, 6. */
/*
/* The default value for FILETYPE_NAME = "Single File Granule" is */
/* usually all that is needed. This means that the input/output only */
/* has one file per granule. Note that this is separate from the */
/* MIN/MAX_GRANULES_REQUIRED and MIN/MAX_GRANULE_YIELD parameters which */
/* tell how many granules are desired for the PCF entry. */
/*
/* If the Data Type defined under this PCF entry can have multiple */
/* files per data granule then this entry must be updated and there has */
/* to be a corresponding entry in the ESDT ODL file for this Data Type. */
/* There needs to be one of these File Type objects for every File Type */
/* associated with this PCF entry. This object defines what file */
/* type(s) this PGE wants to use for this PCF entry. */
/*
/* Note that for L0 inputs, there should only be 1 File Type (different */
/* than "Single File Granule") that defines the number of files in a */
/* L0 granule. */
/*
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
*****/

OBJECT = FILETYPE

/*****
/*      Class (object counter, used only to distinguish objects) */
/*      -- Must be an integer */
/*      -- Must be unique in this file */
/*      Example */
/*      CLASS = 1 */
*****/

CLASS = 1

/*****
/*      Name of File Type. */
/*      -- Must be a string, max len 40 characters. Should */
/*      be meaningful in that the name indicates what sort of */
/*      data is stored within this file type. */
/*      -- Defines what File Type is associated with this PCF */
/*      entry. It will determine how many entries are */
/*      created under this logical ID in the PCF. */
/*      Example */
/*      FILETYPE_NAME = "Instrument Band 7" */
*****/

FILETYPE_NAME = "Single File Granule"

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
*****/

END_OBJECT = FILETYPE

```

```

/*****
/* AUXILIARY LOGICAL ID object */
/*
/* Defines auxiliary logical Ids for a particular input. */
/* This is used when there may be multiple granules for a particular */
/* Logical Id and the PGE wants each granule under a separate logical */
/* Id. The best example of this is the case where a specific L0 */
/* input could have multiple granules satisfying the given time period. */
/* Since only 1 L0 granule is allowed per logical Id, Auxiliary Logical */
/* Ids can be used to spread the subsequent L0 granules among many */
/* Logical IDs. */
/*
/* When Auxiliary Logical Ids are specified, the first granule that */
/* satisfies the input requirements (time period, metadata checks, */
/* etc.) will be placed under the Logical Id defined under the */
/* PCF_ENTRY. Each subsequent granule will be placed under an */
/* Auxiliary Logical Id. The granules are sorted by time, so the */
/* earliest will go under the PCF_ENTRY Logical Id, with the */
/* Auxiliary Logical Ids filled with later and later granules. */
/*
/* There can be more than one AUXILIARY_LOGICAL_ID per PCF_ENTRY, */
/* and if there is one AUXILIARY_LOGICAL_ID object, then there has to */
/* the same number as specified for MAX_GRANULES_REQUIRED. */
/*
/* This object is optional for PCF_ENTRY objects with */
/* PCF_FILE_TYPE = 1, 3 or 6(ignored otherwise). If not needed, this */
/* object should be deleted. */
/*
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****

OBJECT = AUXILIARY_LOGICAL_ID

/*****
/* Class (object counter, used only to distinguish objects) */
/* -- Must be an integer */
/* -- Must be unique in this file for this type of object */
/* -- Must be greater than 0. */
/* Example */
/* CLASS = 1 */
/*****

CLASS = 1

/*****
/* Auxiliary Logical Id */
/* -- The Logical Id to place subsequent granules under */
/* when creating the PCF. */
/* -- Must be a positive integer. */
/* -- The Ids specified for Toolkit use (10000 to 10999) */
/* will not be allowed. */
/* Example: */
/* AUX_LOGICAL_ID = 1001 */
/*****

AUX_LOGICAL_ID =

```



```

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
*****/

    END_OBJECT = AUXILIARY_LOGICAL_ID

/*****
/* Alternate Input object */
/*
/* Defines parameter names and values for this Data Input to be */
/* designated as an "alternate input." This is defined as an input */
/* that can be substituted for another, already defined input. */
/*
/* Note that the "Primary" or first choice Alternate input is also */
/* designated an Alternate input and thus should have one of these */
/* objects. Order should be set to 1. All subsequent Alternates */
/* should have the same Alternate_Category as the primary and should */
/* have Order > 1. */
/*
/* This object is optional for PCF ENTRY objects with */
/* PCF_FILE_TYPE = 1, 3 or 6(ignored otherwise). If not needed, this */
/* object should be deleted. */
/*
/* There can only be one of these objects per PCF ENTRY. */
/*
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
*****/

    OBJECT = ALTERNATE_INPUT

/*****
/*      Class (object counter, used only to distinguish objects) */
/*      -- Must be an integer */
/*      -- Must be unique in this file for this type of object */
/*      -- Must be greater than 0. */
/*      Example */
/*      CLASS = 1 */
*****/

    CLASS = 1

/*****
/*      Name of Alternate Category */
/*      -- Must be a string, max len 20 characters */
/*      -- This is the grouping of Alternates for which this */
/*      entry belongs. The ORDER parameter defines which */
/*      of the alternates is primary, secondary ... */
/*      -- There should be at least one other entry with the */
/*      category. */
/*      Example: */
/*      CATEGORY = "SeaSurfTemp" */
*****/

    CATEGORY = ""

/*****
/*      Default Order for this Alternate */
*/

```

```

/*          Indicates the order of preference for alternates */
/*          within the same category. */
/*          The primary (or first choice alternate) should have */
/*          ORDER = 1. */
/*          -- Must be an integer value. */
/*          -- Should be no greater than the maximum number of */
/*          alternates for the specified CATEGORY. */
/*          Example */
/*          ORDER = 1 (this would be the primary alternate) */
/*****

```

ORDER =

```

/*****
/*          Runtime Parameter Logical Id for this Alternate. */
/*          Sets up a runtime parameter (defined in the User */
/*          Defined Runtime Parameters section of the PCF) that will */
/*          hold the logical ID of the chosen Alternate. */
/*          -- Must be a positive integer value. */
/*          -- Must NOT be a Toolkit specific logical ID */
/*          (10000 and 10999) */
/*          -- Must have a corresponding Runtime Parameter defined */
/*          in PCF section 5. */
/*          Example */
/*          RUNTIME_PARM_ID = 11111 */
/*****

```

RUNTIME_PARM_ID =

```

/*****
/*          Default Timer value to wait for Alternate to be available */
/*          -- Must contain a single P=V string, where */
/*          P is one of { MONTHS, WEEKS, DAYS, HOURS, MINS, SECS} */
/*          -- NOTE that this is not needed if WAITFOR (next */
/*          parameter) is set to "Y". */
/*          Example */
/*          TIMER = "DAYS=1" */
/*****

```

TIMER = "PV_Time_Value_goes_here"

```

/*****
/*          Wait For flag */
/*          Informs PDPS to wait for the alternate input (regardless */
/*          of the timer value). This means that even if the timer */
/*          expires, PDPS will wait for it before executing the */
/*          the PGE. */
/*          -- A character value of either "Y" (YES) or "N" (NO). */
/*          -- Must be set the same for all Alternates in the */
/*          specified CATEGORY. If one Alternate in the CATEGORY */
/*          is set to "Y" then all WAITFOR flags for Alternates */
/*          in that list also must have WAITFOR set to "Y". */
/*          Example */
/*          WAITFOR = "N" */
/*****

```

WAITFOR = ""

```

/*****
/*      Temporal Flag                                */
/*      Indicates if the alternate should be the previous      */
/*      incarnation of the Data Product (Y) rather than the      */
/*      most current Product (N).                                */
/*      -- A character value of either "Y" (YES) or "N" (NO).      */
/*      Example                                                */
/*      TEMPORAL = "N"                                          */
*****/

TEMPORAL = ""

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present      */
*****/

END_OBJECT = ALTERNATE_INPUT

/*****
/* Optional Input object                                          */
/*                                                                */
/* Defines parameter names and values for this Data Input to be */
/* designated as an "optional input." This means that it is an input */
/* that is desired (if available), but that the PGE can process data */
/* successfully without it.                                       */
/*                                                                */
/* Note that Optional Inputs can work like Alternates, in that there */
/* can be a selection to choose from and an order of preference.      */
/* In this case the first choice Optional input would be the "Primary" */
/* (ORDER = 1). If multiple Optional inputs are desired, it is best if */
/* they can be grouped as a list of "Primary" and its "Alternates".    */
/*                                                                */
/* This object is optional for PCF ENTRY objects with              */
/* PCF_FILE_TYPE = 1, 3 or 6 (ignored otherwise).                  */
/*                                                                */
/* There can only be one of these objects per PCF ENTRY.          */
/* An input can either be Alternate or Optional, not both.          */
/* If a PCF entry is not Optional, this object should be deleted.    */
/*                                                                */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present      */
*****/

OBJECT = OPTIONAL_INPUT

/*****
/*      Class (object counter, used only to distinguish objects) */
/*      -- Must be an integer                                */
/*      -- Must be unique in this file for this type of object */
/*      -- Must be greater than 0.                            */
/*      Example                                                */
/*      CLASS = 1                                              */
*****/

CLASS = 1

/*****
/*      Name of Optional Category                                */
*****/

```

```

/*          -- Must be a string, max len 40 characters          */
/*          -- This is the grouping of optional inputs (one or more) */
/*          for which this entry belongs. The ORDER parameter */
/*          defines which of the optionals is primary,          */
/*          secondary ... for the case where there is more than */
/*          one optional input.                                   */
/*          Example:                                           */
/*          CATEGORY = "SeaSurfTemp"                          */
/*****

CATEGORY = ""

/*****
/*          Default Order for this Optional Input          */
/*          Indicates the order of preference for optionals */
/*          within the same category (when there is more than 1). */
/*          The primary (or first choice optional) should have */
/*          ORDER = 1.                                         */
/*          -- Must be an integer value.                      */
/*          -- Should be no greater than the maximum number of */
/*          optionals for the specified CATEGORY.             */
/*          Example                                           */
/*          ORDER = 1 (this would be the primary optional input or */
/*          for a single optional input)                      */
/*****

ORDER =

/*****
/*          Runtime Parameter Logical Id for this Optional Input. */
/*          Sets up a runtime parameter (defined in the User */
/*          Defined Runtime Parameters section of the PCF) that will */
/*          hold the logical ID of the chosen Optional input. */
/*          -- Must be a positive integer value.             */
/*          -- Must NOT be a Toolkit specific logical ID     */
/*          (10000 and 10999)                                 */
/*          -- Must have a corresponding Runtime Parameter defined */
/*          in PCF section 5.                                  */
/*          Example                                           */
/*          RUNTIME_PARM_ID = 11111                          */
/*****

RUNTIME_PARM_ID =

/*****
/*          Default Timer value to wait for Alternate to be available */
/*          -- Must contain a single P=V string, where          */
/*          P is one of { MONTHS, WEEKS, DAYS, HOURS, MINS, SECS} */
/*          -- NOTE that this is not needed if WAITFOR (next */
/*          parameter) is set to "Y".                            */
/*          Example                                           */
/*          TIMER = "DAYS=1"                                   */
/*****

TIMER = "PV_Time_Value_goes_here"

/*****
/*          Temporal Flag                                     */

```

```

/*          Indicates if the alternate should be the previous          */
/*          incarnation of the Data Product (Y) rather than the        */
/*          most current Product (N)                                   */
/*          -- A character value of either "Y" (YES) or "N" (NO).      */
/*          Example                                                    */
/*          TEMPORAL = "N"                                             */
/*****

TEMPORAL = ""

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present          */
/*****

END_OBJECT = OPTIONAL_INPUT

/*****
/* Metadata checks object                                           */
/*                                                                    */
/*          */
/* Defines parameter names, values and conditions for which this PGE */
/* should execute if true for this input file                        */
/* PGE depends.                                                     */
/*                                                                    */
/*          */
/* This object is optional for PCF ENTRY objects with                */
/* PCF_FILE_TYPE = 1,3 or 6 (ignored otherwise). Delete if not needed. */
/* Replicate object if multiple METADATA_CHECKS are required.       */
/*                                                                    */
/*          */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present          */
/*****

OBJECT = METADATA_CHECKS

/*****
/*          Class (object counter, used only to distinguish objects) */
/*          -- This line is generated by DpAtCreateOdlTemplate        */
/*          from the PCF and is normally not modified                */
/*          -- Must be an integer                                     */
/*          -- Must be unique in this file for this type of object   */
/*          -- Must be greater than 0.                                */
/*          Example                                                    */
/*          CLASS = 1                                                  */
/*****

CLASS = 1

/*****
/*          Name of metadata parameter on which this PGE depends     */
/*          -- Must be a string, max len 40 characters.              */
/*          -- Must be present in the ESDT ODL file for this ESDT.   */
/*          -- Means that the specified metadata parameter must have */
/*          the specified value for the PGE to execute.              */
/*          -- For Product Specific Attributes (PSAs), this is the    */
/*          name of the attribute in question. The corresponding      */
/*          entry the ESDT_ODL file must specify CONTAINER_NAME =    */
/*          "AdditionalAttributes".                                    */
/*          Example:                                                  */
/*          The PGE depends on the metadata value for the parameter */
/*          called "tbd_parm_name".                                    */

```

```

/*          PARM_NAME = "tbd_parm_name"          */
/*****

    PARM_NAME = ""

/*****
/*          Operator for dependency condition          */
/*          -- Must be one of { >, <, >=, <=, ==, != }          */
/*          -- This means that the metadata parameter is:          */
/*          ">" -- actual parameter value must be greater than          */
/*                  value specified in VALUE.          */
/*          "<" -- actual parameter value must be less than          */
/*                  value specified in VALUE.          */
/*          ">=" -- actual parameter value must be greater than          */
/*                  or equal to value specified in VALUE.          */
/*          "<=" -- actual parameter value must be less than or          */
/*                  equal to value specified in VALUE.          */
/*          "==" -- actual parameter value must be equal to          */
/*                  value specified in VALUE.          */
/*          "!=" -- actual parameter value must be NOT equal to          */
/*                  value specified in VALUE.          */
/*          Example          */
/*          OPERATOR = "=="          */
/*****

    OPERATOR = ""

/*****
/*          Value for metadata parameter upon which this PGE depends          */
/*          -- The value for the metadata parameter that is to be          */
/*                  checked against.          */
/*          -- Computer data type (string, float or long) of the          */
/*                  value must correspond to the computer data type          */
/*                  given in the ESDT ODL file          */
/*          Example          */
/*          VALUE = 0          */
/*          Requires that TYPE = "INT" for the "tbd_parm_name" object          */
/*          in ODL file          */
/*          $DPAT_ESDT_SCIENCE_MD/ESDT_<ESDTName#Version>.odl          */
/*          VALUE = "Joe"          */
/*          Requires that TYPE = "STR" for the "tbd_parm_name" object          */
/*          in ODL file          */
/*          $DPAT_ESDT_SCIENCE_MD/ESDT_<ESDTName#Version>.odl          */
/*****

    VALUE = ""

/*****
/*          Database query Value          */
/*          -- OPTIONAL parameter. Defaults to "NONE".          */
/*          -- Set to define this Metadata Query as having a          */
/*          a VALUE set by PDPS based on the run of the PGE.          */
/*          This Metadata Query will then be performed on the          */
/*          value retrieved from the PDPS database rather than          */
/*          the value specified in the VALUE parameter.          */
/*          -- Must be one of {"NONE", "PATH NUMBER",          */
/*          "ORBIT NUMBER", "TILE ID", "START DATA DAY",          */
/*          "END DATA DAY", "ORBIT IN DAY", "GRANULE IN ORBIT",          */

```

```

/*      "YEAR OF DATA", "MONTH OF DATA", "DAY OF DATA"}      */
/*      "NONE" -- no dynamic value, use VALUE                  */
/*      "PATH NUMBER" -- get the orbital path number           */
/*      "ORBIT NUMBER" -- get the number of the orbit          */
/*      "TILE ID" -- get the id of the tile                    */
/*      "START DATA DAY" -- get the start data day            */
/*      "END DATA DAY" -- get the end data day                */
/*      "ORBIT IN DAY" -- get the orbit number within day      */
/*      "GRANULE IN ORBIT" -- get the granule within the       */
/*      orbit assuming 6 minute                                */
/*      "YEAR OF DATA" -- the year of the data                */
/*      "MONTH OF DATA" -- the month of the data              */
/*      "DAY OF DATA" -- the day of the data                  */
/*      Example                                                */
/*      DATABASE_QUERY = "PATH NUMBER"                        */
/*****

```

DATABASE_QUERY = "NONE"

```

/*****
/*      Optional Parameter. Defaults to empty string if not specified.      */
/*      */
/*      Name of metadata parameter which provides a key into a      */
/*      a multi-containered object. Such an object is the      */
/*      MeasuredParameters group in the inventory metadata.      */
/*      -- Must be a string, max len 40 characters.              */
/*      -- Must be present in the ESDT ODL file for this ESDT.    */
/*      -- Is matched with KEY_PARAMETER_VALUE to determine      */
/*      the entry in a multi-containered metadata group. */
/*      -- For Product Specific Attributes (PSAs), this entry      */
/*      should NOT be specified. */
/*      -- Because of Metadata Query limitations, there can only      */
/*      be one KEY_PARAMETER_NAME/KEY_PARAMETER_VALUE pair */
/*      per PGE ODL File. This is because only a single */
/*      Metadata Query is allowed against the */
/*      MeasuredParameters group. */
/*      -- For Metadata Queries within the MeasuredParameters      */
/*      group this should be set to the metadata field called      */
/*      "ParameterName". */
/*      Example: */
/*      KEY_PARAMETER_NAME = "ParameterName" */
/*****

```

KEY_PARAMETER_NAME = ""

```

/*****
/*      Optional Parameter. Must be preset if KEY_PARAMETER_NAME exists.      */
/*      Defaults to the empty string if not specified. */
/*      */
/*      Value of metadata parameter which provides a key into a      */
/*      a multi-containered object. Such an object is the      */
/*      MeasuredParameters group in the inventory metadata.      */
/*      -- Must be a string, max len 80 characters.              */
/*      -- Must be present in the ESDT ODL file for this ESDT.    */
/*      -- Is matched with KEY_PARAMETER_NAME to determine      */
/*      the entry in a multi-containered metadata group. */
/*      -- For Product Specific Attributes (PSAs), this entry      */
/*      should NOT be specified. */

```

```

/*      -- Because of Metadata Query limitations, there can only      */
/*      be one KEY_PARAMETER_NAME/KEY_PARAMETER_VALUE pair          */
/*      per PGE ODL File. This is because only a single              */
/*      Metadata Query is allowed against the                        */
/*      MeasuredParameters group.                                    */
/*      -- For Metadata Queries within the MeasuredParameters        */
/*      group this should be set to the desired value of the         */
/*      metadata field called "ParameterName".                        */
/*      Example:                                                      */
/*      KEY_PARAMETER_VALUE = "LandCoverage"                          */
/*****/

KEY_PARAMETER_VALUE = ""

/*****/
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present          */
/*****/

END_OBJECT = METADATA_CHECKS

/*****/
/* Metadata Query Object                                             */
/*                                                                    */
/* Defines parameter names, values and conditions for which this Input */
/* for the PGE should be selected. Only data that matches the         */
/* with the specified metadata parameter with the specified value and  */
/* condition will be chosen as input to this PGE. Note that if no      */
/* matching data is found the PGE will NOT execute.                    */
/*                                                                    */
/* This object is optional for PCF ENTRY objects with                  */
/* PCF_FILE_TYPE = 1,3 or 6 (ignored otherwise). Delete if not needed. */
/* Replicate object if multiple METADATA_QUERYs are required.         */
/*                                                                    */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present          */
/*****/

OBJECT = METADATA_QUERY

/*****/
/* Class (object counter, used only to distinguish objects)          */
/*      -- This line is generated by DpAtCreateOdlTemplate            */
/*      from the PCF and is normally not modified                    */
/*      -- Must be an integer                                          */
/*      -- Must be unique in this file for this type of object       */
/*      -- Must be greater than 0.                                    */
/*      Example                                                        */
/*      CLASS = 1                                                      */
/*****/

CLASS = 1

/*****/
/* Name of metadata parameter on which this PGE depends              */
/*      -- Must be a string, max len 40 characters                    */
/*      -- Must be present in the ESDT ODL file for this ESDT        */
/*      Example:                                                       */
/*      This CERES PGE depends on the Q/A value of                    */
/*      this ESDT "TRWpcall182": execute the CERES PGE only          */

```



```

/*          if ESDT "TRWpcall182" had Q/A parameter          */
/*          "tbd_parm_name" = 0                               */
/*          PARM_NAME = "tbd_parm_name"                       */
/*****

    PARM_NAME = "Parm_name_goes_here"

/*****
/*          Operator for dependency condition                    */
/*          -- Must be one of { >, <, >=, <=, ==, != }          */
/*          Example                                              */
/*          OPERATOR = "=="                                       */
/*****

    OPERATOR = "Operator_goes_here"

/*****
/*          Value for ESDT parameter upon which this PGE depends */
/*          -- Computer data type (string, float or long) of the */
/*          value must correspond to the computer data type */
/*          given in the ESDT ODL file                            */
/*          Example                                              */
/*          VALUE = 0                                           */
/*          Requires that TYPE = "INT" for the "tbd_parm_name" object */
/*          in ODL file                                         */
/*          $DPAT_ESDT_SCIENCE_MD/ESDT_<ESDTName#Version>.odl */
/*          VALUE = "Joe"                                       */
/*          Requires that TYPE = "STR" for the "tbd_parm_name" object */
/*          in ODL file                                         */
/*          $DPAT_ESDT_SCIENCE_MD/ESDT_<ESDTName#Version>.odl */
/*****

    VALUE = "Value_goes_here"

/*****
/*          Database query Value                                */
/*          -- OPTIONAL parameter. Defaults to "NONE".          */
/*          -- Set to define this Metadata Query as having a    */
/*          a VALUE set by PDPS based on the run of the PGE.    */
/*          This Metadata Query will then be performed on the   */
/*          value retrieved from the PDPS database rather than  */
/*          the value specified in the VALUE parameter.         */
/*          -- Must be one of {"NONE", "PATH NUMBER",          */
/*          "ORBIT NUMBER", "TILE ID", "START DATA DAY",      */
/*          "END DATA DAY", "ORBIT IN DAY", "GRANULE IN ORBIT", */
/*          "YEAR OF DATA", "MONTH OF DATA", "DAY OF DATA"}  */
/*          "NONE" -- no dynamic value, use VALUE              */
/*          "PATH NUMBER" -- get the orbital path number        */
/*          "ORBIT NUMBER" -- get the number of the orbit       */
/*          "TILE ID" -- get the id of the tile                 */
/*          "START DATA DAY" -- get the start data day         */
/*          "END DATA DAY" -- get the end data day             */
/*          "ORBIT IN DAY" -- get the orbit number within day   */
/*          "GRANULE IN ORBIT" -- get the granule within the    */
/*          orbit assuming 6 minute                             */
/*          "YEAR OF DATA" -- the year of the data             */
/*          "MONTH OF DATA" -- the month of the data           */
/*          Example                                              */

```

```

/*          DATABASE_QUERY = "PATH NUMBER"                                     */
/*****

    DATABASE_QUERY = "NONE"

/*****
/* Optional Parameter. Defaults to empty string if not specified.             */
/*                                     */
/*      Name of metadata parameter which provides a key into a                */
/*      a multi-containered object. Such an object is the                     */
/*      MeasuredParameters group in the inventory metadata.                   */
/*      -- Must be a string, max len 40 characters.                           */
/*      -- Must be present in the ESDT ODL file for this ESDT.               */
/*      -- Is matched with KEY_PARAMETER_VALUE to determine                   */
/*      the entry in a multi-containered metadata group.                      */
/*      -- For Product Specific Attributes (PSAs), this entry                 */
/*      should NOT be specified.                                               */
/*      -- For Metadata Checks within the MeasuredParameters                 */
/*      group this should be set to the metadata field called                 */
/*      "ParameterName".                                                       */
/*      Example:                                                                */
/*      KEY_PARAMETER_NAME = "ParameterName"                                   */
/*****

    KEY_PARAMETER_NAME = ""

/*****
/* Optional Parameter. Must be preset if KEY_PARAMETER_NAME exists.           */
/* Defaults to the empty string if not specified.                             */
/*                                     */
/*      Value of metadata parameter which provides a key into a                */
/*      a multi-containered object. Such an object is the                     */
/*      MeasuredParameters group in the inventory metadata.                   */
/*      -- Must be a string, max len 80 characters.                           */
/*      -- Must be present in the ESDT ODL file for this ESDT.               */
/*      -- Is matched with KEY_PARAMETER_NAME to determine                   */
/*      the entry in a multi-containered metadata group.                      */
/*      -- For Product Specific Attributes (PSAs), this entry                 */
/*      should NOT be specified.                                               */
/*      -- For Metadata Checks within the MeasuredParameters                 */
/*      group this should be set to the desired value of the                 */
/*      metadata field called "ParameterName".                                */
/*      Example:                                                                */
/*      KEY_PARAMETER_VALUE = "LandCoverage"                                   */
/*****

    KEY_PARAMETER_VALUE = ""

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present                  */
/*****

    END_OBJECT = METADATA_QUERY

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present                  */
/*****

```

```
END_OBJECT = PCF_ENTRY
```

```
/* After this point, the comments only address unique parameters that
/* have not been explained above
/*
/* Note that the order of PCF entries is not really important. These
/* have been ordered the same as their order would be in the PGEs PCF.
/*
```

```
OBJECT = PCF_ENTRY
CLASS = 2
LOGICAL_ID = 3000
PCF_FILE_TYPE = 2
DATA_TYPE_NAME = ""
```

```
/* Minimum number of output granules for this logical ID
/* -- This line is generated by DpAtCreateOdlTemplate
/* from the PCF and is only modified if the PGE may
/* successfully produce less granules than specified in
/* the PCF used to generate the template.
/* -- Required for all PCF ENTRY objects with
/* PCF_FILE_TYPE = 2, 4, 7 (ignored otherwise).
/* -- Must be a positive integer.
/* -- Note that for number of files within a granule
/* greater than one, the FILE TYPE object for this entry
/* must be changed to specify the various file types and
/* maximum number of files.
/* Example
/* MIN GRANULE YIELD = 1
```

```
MIN_GRANULE_YIELD = 1
```

```
/* Maximum number of output granules for this logical ID
/* -- This line is generated by DpAtCreateOdlTemplate
/* from the PCF and is only modified if the PGE may
/* successfully produce more granules than specified in
/* the PCF used to generate the template.
/* -- Required for all PCF ENTRY objects with
/* PCF_FILE_TYPE = 2, 4, 7 (ignored otherwise).
/* -- Must be a positive integer.
/* -- Note that for number of files within a granule
/* greater than one, the FILE TYPE object for this entry
/* must be changed to specify the various file types and
/* maximum number of files.
/* Example
/* MAX GRANULE YIELD = 1
```

```
MAX_GRANULE_YIELD = 1
```

```
/* Associated MCF ID
```

```

/*          -- The Logical ID of the MCF associated with this input. */
/*          Informs Data Processing as to the logical id which */
/*          the PGE associates the MCF for this output. */
/*          -- Required for all PCF ENTRY objects with */
/*          PCF_FILE_TYPE = 2, 4 (if not output by the Toolkit), */
/*          7. (ignored otherwise). */
/*          -- Must be a positive integer. */
/*          -- NOTE that any input PCF entries that were created */
/*          by CreateOdlTemplate for MCFs should be deleted. The */
/*          information about which Logical IDs are for MCFs is */
/*          is captured by this parameter for each output that */
/*          the MCF is associated with. */
/*          Example */
/*          ASSOCIATED_MCF_ID = 3001
/*****

ASSOCIATED_MCF_ID =

/*****
/*          Output file group ID */
/*          -- Required for all PCF ENTRY objects with */
/*          PCF_FILE_TYPE = 2 (ignored otherwise) */
/*          -- Must be a string */
/*          -- 1st character must be one of {S,Q,H,B} */
/*          S -- Science file */
/*          Q -- Q/A file */
/*          H -- Production history file */
/*          B -- Browse file */
/*          -- Rest of string must resolve to a */
/*          positive integer < 1000 */
/*          Example */
/*          SCIENCE_GROUP = "S1" */
/*          Files associated with this science file would have */
/*          SCIENCE_GROUP = "Q1", SCIENCE_GROUP = "B1", etc. */
/*****

SCIENCE_GROUP = " "

/*****
/*          Nominal no. of file instances with *different* logical IDs, */
/*          but which are associated with each other */
/*          -- Optional for all PCF ENTRY objects with */
/*          PCF_FILE_TYPE = 2 (ignored otherwise). */
/*          -- If 0, ignore this parameter -- no other logical IDs */
/*          are associated with it. */
/*          Example */
/*          INSTANCE = 0 */
/*          Note: This parameter is specifically designed to accomodate */
/*          the CERES case where 24 standard product files are generated */
/*          per day, each with a *different* logical ID, but are all */
/*          essentially an instance of a single file format */
/*          In this case INSTANCE would take values 1, 2, ..., 24 */
/*****

INSTANCE = 0

/*****
/*          Distinct Value for the output. */

```

```

/*          -- Optional entry for PCF ENTRY objects with          */
/*          PCF_FILE_TYPE = 2,4,7.  Set to null if not provided.  */
/*          -- A string value, max length 80 characters.          */
/*          -- A value that will allow unique naming of granules  */
/*          produced by a PGE.                                     */
/*          -- Must be the name of a metadata parameter defined in */
/*          a METADATA_DEFINITION object.  If a parameter is      */
/*          is specified for which no METADATA_DEFINITION object  */
/*          exists an error will be raised during ODL parsing.    */
/*          -- Supports what are called Multi-Granule ESDTs. These */
/*          are ESDTs that have multiple granules for the same    */
/*          time period where the only difference between the     */
/*          granules is metadata parameters.                      */
/*          Example                                               */
/*          DISTINCT_VALUE = "CAMERA_DF"                          */
/*****

DISTINCT_VALUE = ""

/*****
/*          Minimum expected size (in MB) of this output          */
/*          (used for QA purposes).                                */
/*          -- Required for all PCF ENTRY objects with            */
/*          PCF_FILE_TYPE = 2 (ignored otherwise)                  */
/*          -- Must be a positive integer                          */
/*          Example                                               */
/*          MINIMUM_SIZE = 120000                                  */
/*****

MINIMUM_SIZE = 0

/*****
/*          Maximum expected size (in MB) of this output          */
/*          (used for QA purposes).                                */
/*          -- Required for all PCF ENTRY objects with            */
/*          PCF_FILE_TYPE = 2 (ignored otherwise)                  */
/*          -- Must be a positive integer                          */
/*          -- Must be larger than or equal to MINIMUM_SIZE       */
/*          Example                                               */
/*          MAXIMUM_SIZE = 50000000                               */
/*****

MAXIMUM_SIZE = 1

OBJECT = FILETYPE
CLASS = 1
FILETYPE_NAME = "Single File Granule"
END_OBJECT = FILETYPE

/*****
/* Associated Science Data Object                                */
/*          */
/* THIS OBJECT IS REQUIRED for Outputs where the SCIENCE_GROUP    */
/* contains 'B' or 'Q' (meaning it is a BROWSE or QA granule).  It is */
/* ignored otherwise.                                             */
/*          */
/* BROWSE and QA output granules are linked to the science granules */
/* for which they are produced.  This linkage occurs when the produced */

```

```

/* BROWSE or QA granules are inserted to the Data Server. This object */
/* defines the linkage so that the correct link can be made after */
/* the PGE completes and its outputs are inserted to the Data Server. */
/* */
/* If more than one science granule is associated with the BROWSE or */
/* QA output defined by this PCF_ENTRY, then repeat the Associated */
/* Science Data Objects to specify the various Logical Ids that define */
/* those Associated Science Granules. */
/* */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****

OBJECT = ASSOCIATED_SCIENCE_DATA

/*****
/*      Class (object counter, used only to distinguish objects) */
/*      -- Must be an integer */
/*      -- Must be unique in this file */
/*      Example */
/*      CLASS = 1 */
/*****

CLASS = 1

/*****
/*      Associated Science Granule's Logical Id */
/*      -- Must a positive integer value. */
/*      -- Defines which logical Id this BROWSE/QA granules is */
/*      Associated with. This means that when the associated */
/*      science granule is inserted to the Data Server, a */
/*      will be made with the BROWSE/QA granule defined by */
/*      this PCF_ENTRY. */
/*      -- A check will be done to verify that the Logical ID */
/*      has been defined in the ODL file. */
/*      Example */
/*      ASSOCIATED_SCIENCE_LOGICAL_ID = 12345 */
/*****

ASSOCIATED_SCIENCE_LOGICAL_ID =

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****

END_OBJECT = ASSOCIATED_SCIENCE_DATA

END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY

CLASS = 3
LOGICAL_ID = 200

/*****
/* This is an example for in Support input. */
/*****
PCF_FILE_TYPE = 3

```

```

/*****
/* Support input and output types (if not associated with generic      */
/* Toolkit files) have their own Data Types and Versions.              */
/*****

    DATA_TYPE_NAME = ""
    DATA_TYPE_VERSION = ""

/*****
/* This is always 1 for non-product inputs                             */
/*****
    DATA_TYPE_REQUIREMENT = 1

/*****
/* Support inputs can be any input type.  Though none are             */
/* shown, they can have Alternate or Optional input objects as well    */
/* Metadata checks objects.                                             */
/*****
    INPUT_TYPE = ""
    NUMBER_NEEDED = 1

    OBJECT = FILETYPE
        CLASS = 1
        FILETYPE_NAME = "Single File Granule"
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY

    CLASS = 4
    LOGICAL_ID = 4000
    PCF_FILE_TYPE = 5

/*****
/*      Runtime parameter name                                         */
/*      -- This line is generated by DpAtCreateOdlTemplate            */
/*      from the PCF and is normally not modified                    */
/*      -- Required for all PCF ENTRY objects with                    */
/*      PCF_FILE_TYPE = 5 (ignored otherwise)                          */
/*      -- Must be a string, max len 50 characters                    */
/*      Example                                                         */
/*      PGE_PARAMETER_NAME = "Spacecraft_Class"                       */
/*****

    PGE_PARAMETER_NAME = " "

/*****
/*      Runtime parameter default value                                */
/*      -- This line is generated by DpAtCreateOdlTemplate            */
/*      from the PCF and is normally not modified                    */
/*      -- Required for all PCF ENTRY objects with                    */
/*      PCF_FILE_TYPE = 5 (ignored otherwise)                          */
/*      -- Must be a string, max len 200 characters                    */
/*      -- If double quotes must be included in the string            */
/*      (i.e. the string must read "This is the string")              */
/*      then single quotes must be placed around the string.          */
/*      Thus "This is the string" would become '"This is the         */
/*      string"'.  Note that this automatically done by               */

```

```

/*          the CreateODLTemplate tool.          */
/*          Example          */
/*          PGE_PARAMETER_DEFAULT = "TRMM"          */
/*****

PGE_PARAMETER_DEFAULT = " "

/*****
/*          Runtime parameter Dynamic Value          */
/*          -- This line is generated by DpAtCreateOdlTemplate */
/*          from the PCF and is set to "NONE".          */
/*          -- Set to define this runtime parameter as having a */
/*          a value set by PDPS based on the run of the PGE. */
/*          This runtime parameter will then have the value of */
/*          the specified attribute when the PCF is created. */
/*          -- Required for all PCF ENTRY objects with */
/*          PCF_FILE_TYPE = 5 (ignored otherwise)          */
/*          -- Must be one of {"NONE", "PATH NUMBER", */
/*          "ORBIT NUMBER", "TILE ID", "START DATA DAY", */
/*          "END DATA DAY"}          */
/*          "NONE" -- no dynamic value, use Default */
/*          "PATH NUMBER" -- get the orbital path number */
/*          "ORBIT NUMBER" -- get the number of the orbit */
/*          "TILE ID" -- get the id of the tile */
/*          "START DATA DAY" -- get the start data day */
/*          "END DATA DAY" -- get the end data day */
/*          Example          */
/*          PGE_PARAMETER_DYNAMIC_VALUE = "PATH NUMBER"          */
/*****

PGE_PARAMETER_DYNAMIC_VALUE = "NONE"

/*****
/*          Profile Selector Runtime Parameter Flag          */
/*          -- This line is generated by DpAtCreateOdlTemplate */
/*          from the PCF and is set to "N".          */
/*          -- Must be a string, value of either "Y" (for Yes) and */
/*          "N" (for No).          */
/*          -- If not specified, defaults to "N".          */
/*          -- Indicates that this Runtime Parameter (along with */
/*          others) uniquely identifies a profile of this PGE */
/*          (PGE Name + PGE version) based on the PARAMETER_NAME */
/*          and DEFAULT_VALUE pair.          */
/*          -- If set to "Y" for any Runtime Parameter, then the */
/*          RegisterPGE tool will check to make sure that this */
/*          Runtime Parameter/Default Value pairs flagged */
/*          assures that this PGE Profile is different from all */
/*          the rest.          */
/*          Example          */
/*          PROFILE_SELECTOR_PGE_PARAMETER = "N"          */
/*****

PROFILE_SELECTOR_PGE_PARAMETER = ""

END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY

```



```

CLASS = 5
LOGICAL_ID = 200

/*****
/* This is an example for an Interim/Intermediate input.          */
*****/
PCF_FILE_TYPE = 6

/*****
/* Interim/Intermediate input and output types have their own Data */
/* Types and Versions.                                           */
*****/

DATA_TYPE_NAME = ""
DATA_TYPE_VERSION = ""

/*****
/*      Last PGE to Use Interim Data Type?                        */
/*      This is a "Y" or "N" parameter that defines if this PGE   */
/*      (the one defined by this ODL file) is the last to use this */
/*      Interim Data type.                                         */
/*      -- Must be a string or "Y" or "N".                         */
/*      Example                                                    */
/*      INTERIM_LAST_PGE_TO_USE = "N"                             */
*****/

INTERIM_LAST_PGE_TO_USE = "N"

DATA_TYPE_REQUIREMENT = 1

/*****
/* Interim/Intermediate inputs can be any input type.  Though none are */
/* are shown, they can have Alternate or Optional input objects as well */
/* Metadata checks objects.                                           */
*****/
INPUT_TYPE = ""
NUMBER_NEEDED = 1

OBJECT = FILETYPE
CLASS = 1
FILETYPE_NAME = "Single File Granule"
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED                          */
*****/

END

```

A.1.2 ESDT_ODL.template

```

/*****
/*
/*      TEMPLATE ESDT SCIENCE METADATA ODL FILE
/*
*****/

```

```

/*                                                    */
/* The SSIT operator's responsibility is to copy this file over and */
/* edit it to add all necessary PDPS metadata values.                */
/*                                                    */
/* Each ESDT used by a PGE must have a corresponding ESDT SCIENCE */
/* metadata ODL file.                                                */
/*                                                    */
/* All ESDT ODL files must reside in directory $DPAT_ESDT_SCIENCE_MD . */
/*                                                    */
/* The operator must add a value to the right of the "=" for each */
/* parameter.                                                        */
/*                                                    */
/* CHANGE LOG                                                        */
/* -- Added File Type object.                                05/28/97 */
/* -- Added Processing Level.                                06/04/97 */
/* -- Added Orbit types to period/boundary comments.        06/07/97 */
/* -- Updated Archived_By and Processed_By to be            */
/*     required for all types but Static.                    06/24/97 */
/* -- Allowed for 0 value in Interim Short Delete.          10/07/97 */
/* -- Added DURATION parameter.                              11/14/97 */
/* -- Removed OVERLAP as a choice for prediction meth.     11/03/97 */
/* -- Changed METADATA_CHECKS to METADATA_DEFINITION.       12/06/97 */
/*     Updated description of FILETYPE object.              */
/* -- Added optional METADATA_CONTAINER parameter.          12/15/97 */
/* -- Added info on Metadata_Definition for                 02/04/98 */
/*     Product Specific Attributes.                          */
/* -- Added The Distinct Parameter definition.              03/24/98 */
/* -- Fixed length for PROVIDER, FILETYPE_NAME.            03/31/98 */
/* -- Made CONTAINER no longer optional for METADATA       05/06/98 */
/*     DEFINITION object.                                    */
/* -- Updated definition of USE_OBJECT.                      06/25/98 */
/* -- Added KEY_PARAMATER_NAME and KEY_PARAMETER_VALUE     07/05/98 */
/*     for Metadata Definition objects.                      */
/*     Updated description for DISTINCT_PARAMETER.          */
/* -- Updated lengths for INSTRUMENT and PLATFORM.         08/13/98 */
/* -- Added "NONROUTINE" for PREDICTION_METHOD             08/24/98 */
/*     parameter. This is for ASTER Routine Processing.     */
/* -- Added PRODUCTION_CHAIN object                         07/12/99 */
/* -- Added ONDEMAND_DELETION_INTERVAL parameter           */
/*                                                    */
/*****/

/*****/
/* Data Type name */
/* -- Must be a string, max len 8 characters */
/* -- ESDT name inside ODL file must be identical to */
/*     ESDT name used as part of ODL filename, */
/*     which in turn was generated from the */
/*     DATA_TYPE_NAME in the PGE ODL file for the PCF */
/*     entry. */
/* -- It should be the same as the Short Name used in the */
/*     ESDT defintion at the Data Server. */
/* Example */
/* DATA_TYPE_NAME = "NMC" */
/*****/

DATA_TYPE_NAME = ""

```

```

/*****
/*      Data Type Version                                */
/*      -- Must be a string, max len 5 characters        */
/*      -- ESDT name inside ODL file must be identical to */
/*      ESDT name used as part of ODL filename,          */
/*      which in turn was generated from the             */
/*      DATA_TYPE_VERSION in the PGE ODL file for the PCF */
/*      entry.                                           */
/*      -- It should be the same as the VersionID used in the */
/*      ESDT definition at the Data Server.              */
/*      -- Note that this is not important for Interim/    */
/*      Intermediate types.                               */
/*      Example                                           */
/*      DATA_TYPE_VERSION = "3.5.1"                    */
*****/

DATA_TYPE_VERSION = ""

/*****
/*      Science instrument name                          */
/*      -- Must be a string, max 20 len characters        */
/*      Example                                           */
/*      INSTRUMENT = "NMC"                                */
*****/

INSTRUMENT = ""

/*****
/*      Spacecraft platform name                          */
/*      -- Must be a string, max len 25 characters        */
/*      Example                                           */
/*      PLATFORM = "NOAA9"                                */
*****/

PLATFORM = ""

/*****
/*      ESDT description                                  */
/*      -- Must be a string, max len 60 characters        */
/*      Example                                           */
/*      DATA_TYPE_DESCRIPTION = "NMC 12-hour forecast"    */
*****/

DATA_TYPE_DESCRIPTION = ""

/*****
/*      ESDT data provider (DAAC name to which files are delivered) */
/*      -- Must be a string, max len 50 characters        */
/*      Example                                           */
/*      PROVIDER = "National Meteorological Center"        */
*****/

PROVIDER = ""

/*****
/*      Nominal ESDT file size in MB                      */
/*      -- Must be a floating point number (i.e., include a ".") */
/*      -- Must be greater than 0.000001                 */
*****/

```

```

/*          Example                                     */
/*          NOMINAL_SIZE = 1.5                         */
/*****/

NOMINAL_SIZE =

/*****/
/*          Processing Level                           */
/*          -- A string defining the level of processing for this */
/*          ESDT.                                       */
/*          -- Must be a string of no more than 6 characters. */
/*          Example                                     */
/*          PROCESSING_LEVEL = "L0"                     */
/*****/

PROCESSING_LEVEL = ""

/*****/
/*          HDF Data Flag                             */
/*          Informs DPS that the data within this ESDT will be */
/*          HDF data (if set to Y). Needed for DPS to correctly */
/*          set the PCF entries for metadata access.          */
/*          -- A character value of either "Y" (YES) or "N" (NO). */
/*          -- This will tell the Toolkit whether to get the */
/*          metadata information from the HDF file of the ASCII */
/*          metadata file.                                  */
/*          Example                                       */
/*          HDF_DATA = "N"                               */
/*****/

HDF_DATA = ""

/*****/
/* THIS PARAMETER IS ONLY REQUIRED FOR files in the INPUT sections */
/* of the PCF (PRODUCT, SUPPORT or INTERMEDIATE)                  */
/* (ignored for output files, which are always type "I")          */
/*                                                                */
/*          Dynamic flag -- flags whether an ESDT is dynamic      */
/*          -- Allowed values:                                     */
/*          "S" -- Static file                                     */
/*          "I" -- Dynamic internal file                         */
/*          "E" -- Dynamic external file                       */
/*          "T" -- Interim/Intermediate file                    */
/*          Example                                             */
/*          DYNAMIC_FLAG = "E"                                   */
/*****/

DYNAMIC_FLAG = ""

/*****/
/* THIS PARAMETER IS ONLY REQUIRED FOR Interim/Intermediate files */
/* (DYNAMIC_FLAG = "T")                                           */
/*                                                                */
/*          Long Duration of Interim/Intermediate files of the */
/*          ESDT before they are be deleted (because no longer needed). */
/*          -- Must be a positive number (0 is NOT allowed).      */
/*          -- Time is specified in MINUTES.                      */
/*          -- This value should be long enough such that there is */

```

```

/*          no chance that the file will be needed at the end of */
/*          this duration.                                         */
/*          Example                                                */
/*          INTERIM_LONG_DURATION = 7200 (5 days)                  */
/*****

INTERIM_LONG_DURATION =

/*****
/* THIS PARAMETER IS ONLY REQUIRED FOR Interim/Intermediate files */
/* (DYNAMIC_FLAG = "T")                                         */
/*          */
/*          Short Duration of Interim/Intermediate files of the */
/*          ESDT before they are be deleted (because no longer needed). */
/*          -- Must be greater than or equal to 0. It should only */
/*          0 if no other PGE uses this Interim file (i.e. an */
/*          Interim file that a PGE uses internally between */
/*          Processes).                                         */
/*          -- Time is specified in MINUTES.                    */
/*          -- This value is a guess at the soonest (after use and */
/*          any QA checks) at when the Interim File can be */
/*          deleted.                                           */
/*          Example                                             */
/*          INTERIM_SHORT_DURATION = 1440 (24 Hours)            */
/*****

INTERIM_SHORT_DURATION =

/*****
/* THIS PARAMETER IS ONLY REQUIRED FOR Dynamic Internal files */
/* (DYNAMIC_FLAG = "I")                                         */
/*          */
/*          On Demand Deletion Interval. This is the time between */
/*          creation of a granule of this ESDT via an On Demand request */
/*          and when this granule is deleted (because it has been */
/*          distributed to the requestor).                      */
/*          -- Must contain a single P=V string, where          */
/*          P is one of { YEARS, MONTHS, THIRDS, WEEKS, DAYS, */
/*          HOURS, MINS, SECS, ORBITS}                          */
/*          -- Must be greater than or equal to 1 week ("WEEKS=1"). */
/*          An error will be returned if the value specified */
/*          is less than 1 week.                                */
/*          -- If not specified then the value defaults to 1 week */
/*          ("WEEKS=1").                                         */
/*          Example                                             */
/*          ONDEMAND_DELETION_DURATION = "WEEKS=1"              */
/*****

ONDEMAND_DELETION_DURATION = ""

/*****
/* THIS PARAMETER IS ONLY REQUIRED FOR Dynamic External files */
/* (DYNAMIC_FLAG = "E")                                         */
/*          */
/*          Data availability prediction method                  */
/*          -- Must be one of {"ROUTINE", "NONROUTINE"}        */
/*          -- "ROUTINE" = data is expected at regular intervals. */
/*          "NONROUTINE" = data comes in sparatically.          */

```

```

/*          No Period, Boundary or Duration is      */
/*          required for NONROUTINE data.           */
/*          Example                                 */
/*          PREDICTION_METHOD = "ROUTINE"          */
/*****/

PREDICTION_METHOD = ""

/*****/
/* THIS PARAMETER IS ONLY REQUIRED FOR Dynamic External files */
/* (DYNAMIC_FLAG = "E")                                     */
/*                                                         */
/*          Supplier name                                 */
/*          -- Must be a string, max len 30 characters    */
/*          Example                                       */
/*          SUPPLIER_NAME = "NOAA"                      */
/*****/

SUPPLIER_NAME = ""

/*****/
/* THIS PARAMETER IS ONLY REQUIRED FOR Dynamic External files */
/* (DYNAMIC_FLAG = "E")                                     */
/*                                                         */
/*          Nominal collection period within granule      */
/*          -- Must contain a single P=V string, where    */
/*          P is one of { YEARS, MONTHS, THIRDS, WEEKS, DAYS, */
/*          HOURS, MINS, SECS, ORBITS}                    */
/*          -- NOTE that if ORBITS are used PROCESSING_BOUNDARY */
/*          must be set to "START_OF_ORBIT".              */
/*          -- This value is ignored for PREDICTION_METHOD = */
/*          "NONROUTINE"                                   */
/*          Example                                       */
/*          PERIOD = "HOURS=12"                          */
/*****/

PERIOD = ""

/*****/
/* THIS PARAMETER IS ONLY REQUIRED FOR Dynamic External files */
/* (DYNAMIC_FLAG = "E")                                     */
/*                                                         */
/*          Nominal time boundary on which ESDT arrives    */
/*          -- Must contain 1 or more P=V strings, where P is one of */
/*          { START_OF_HOUR, START_OF_6HOUR, START_OF_DAY, */
/*          START_OF_WEEK, START_OF_ONE_THIRD_MONTH, */
/*          START_OF_MONTH, START_OF_YEAR, START_DATE, */
/*          START_OF_ORBIT };                               */
/*          also, "+<n>" or "-<n>" may be added to any of these, */
/*          where <n> specifies integer seconds.           */
/*          For START_DATE an "=" can be added followed by the */
/*          start date.                                     */
/*          -- NOTE that START_OF_ORBIT must be used for Data based */
/*          on an Orbit Model. A file of named              */
/*          ORBIT_<platform>.odl must be present.          */
/*          -- This value is ignored for PREDICTION_METHOD = */
/*          "NONROUTINE"                                   */
/*          Example                                       */

```

```

/*          BOUNDARY      = "START_OF_DAY+10800"          */
/*****

BOUNDARY = ""

/*****
/* OPTIONAL PARAMETER          */
/* ONLY USED FOR Dynamic External files          */
/* (DYNAMIC_FLAG = "E")          */
/*          */
/*          Duration of the data.          */
/*          -- Defines the length of time covered by the data.          */
/*          -- Only needed if length of time covered by the data          */
/*              differs from the value specified in PERIOD.          */
/*          -- Must contain a single P=V string, where          */
/*              P is one of { YEARS, MONTHS, THIRDS, WEEKS, DAYS,          */
/*                  HOURS, MINS, SECS, ORBITS}          */
/*          -- NOTE that if ORBITS are used PROCESSING_BOUNDARY          */
/*              must be set to "START_OF_ORBIT".          */
/*          -- This value is ignored for PREDICTION_METHOD =          */
/*              "NONROUTINE"          */
/*          Example          */
/*          DURATION = "HOURS=12"          */
/*****

DURATION = ""

/*****
/* THIS PARAMETER IS ONLY REQUIRED FOR Dynamic External files          */
/* (DYNAMIC_FLAG = "E")          */
/*          */
/*          Avg delay between granule collection and arrival, in secs          */
/*          -- Must be a positive integer          */
/*          Example          */
/*          DELAY = 43200          */
/*****

DELAY =

/*****
/*          Spatial characteristics of the Data Type.          */
/*          -- Must be a character, "Y" = Yes, spacial          */
/*              characteristics exist, "N" = No, spacial          */
/*              characteristics do not exist.          */
/*          Example          */
/*          SPATIAL_FLAG = "Y"          */
/*****

SPATIAL_FLAG = ""

/*****
/* OPTIONAL parameter          */
/*          Distinct Parameter for Granule naming          */
/*          -- A String, max length 80 characters.          */
/*          -- A value that will allow unique naming of granules          */
/*              produced by a PGE.          */
/*          -- Must be the name of a metadata parameter defined in          */
/*              a METADATA_DEFINITION objected. If a parameter is          */

```

```

/*          is specified for which no METADATA_DEFINITION object          */
/*          exists an error will be raised during ODL parsing.          */
/*          -- Supports what are called Multi-Granule ESDTs. These      */
/*          are ESDTs that have multiple granules for the same          */
/*          time period where the only difference between the            */
/*          granules is metadata parameters.                             */
/*          -- NOTE that this parameter must be unique without           */
/*          including KEY_PARAMETER_NAME and KEY_PARAMETER_VALUE.        */
/*          If the parameter requires it, then they must still be        */
/*          specified, but the value specified for                        */
/*          DISTINCT_PARAMETER cannot need them to be considered         */
/*          unique.                                                        */
/*          Example                                                         */
/*          DISTINCT_PARAMETER = "CAMERA"                                  */
/*****/

DISTINCT_PARAMETER = ""

/*****/
/* Use object                                                                */
/*                                                                    */
/* Defines the DAAC(s) where the data is used.                            */
/*                                                                    */
/* There should be one of these for every DAAC where the data type is    */
/* used. Delete or replicate this object as necessary.                    */
/*                                                                    */
/* This object is really only required for data that is used at a DAAC    */
/* other than where it's produced.                                         */
/*                                                                    */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present              */
/*****/

OBJECT = USE_OBJECT

/*****/
/*          Class (object counter, used only to distinguish objects)      */
/*          -- Must be an integer                                           */
/*          -- Must be unique in this file                                  */
/*          Example                                                         */
/*          CLASS = 1                                                       */
/*****/

CLASS = 1

/*****/
/*          DAAC where the Data Type is used.                             */
/*          -- Must be a string, max len 4 characters. Use the            */
/*          DAAC abbreviation (i.e. GSFC)                                  */
/*          -- There should be one of these for every DAAC where          */
/*          the data type is used.                                         */
/*          Example                                                         */
/*          USED_BY = "GSFC"                                                */
/*****/

USED_BY = ""

/*****/
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present              */

```



```

/*****/

END_OBJECT = USE_OBJECT

/*****/
/* THIS PARAMETER IS REQUIRED FOR ALL types of file but STATIC (S) */
/* (DYNAMIC_FLAG = "S") */
/* */
/* DAAC where the Data Type is archived. */
/* -- Must be a string, max len 4 characters. Use the */
/* DAAC abbreviation (i.e. GSFC) */
/* Example */
/* ARCHIVED_AT = "GSFC" */
/*****/

ARCHIVED_AT = ""

/*****/
/* THIS PARAMETER IS ONLY REQUIRED FOR ALL types of file but STATIC (S) */
/* (DYNAMIC_FLAG = "S") */
/* */
/* DAAC where the Data Type is processed. */
/* -- Must be a string, max len 4 characters. Use the */
/* DAAC abbreviation (i.e. GSFC) */
/* Example */
/* PROCESSED_AT = "GSFC" */
/*****/

PROCESSED_AT = ""

/*****/
/* File Types Object */
/* */
/* THIS OBJECT IS REQUIRED FOR all ESDTs that can have multiple files */
/* per data granule. It is NOT needed for ESDTs where each file */
/* represents a single granule (those inputs in the PGE ODL file that */
/* have "Single File Granule" for the File Type). */
/* */
/* It is up to the PGE writer to determine if multiple files (whether */
/* different types or multiple files for the same type) are */
/* read/written by the PGE. Files and granules differ because a */
/* a granule is the smallest amount of data recognized by the system, */
/* but one granule may be made up of several files. These files */
/* may be of different types, so that only specific information */
/* (specific files) can be requested as input. */
/* */
/* Defines the types of files and their maximum numbers that can be */
/* associated with this ESDT. */
/* */
/* There should be one of these for every File Type that can be */
/* associated with this ESDT. */
/* */
/* Note that this does NOT need to be added for L0 data. Though */
/* such granules are multi-file, they are handled differently by */
/* PDPS. There does not need to be a FILETYPE object in the ESDT ODL */
/* for L0 data. */
/* */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */

```

```

/*****/

OBJECT = FILETYPE

/*****/
/*      Class (object counter, used only to distinguish objects)      */
/*      -- Must be an integer                                          */
/*      -- Must be unique in this file                                */
/*      Example                                                         */
/*      CLASS = 1                                                       */
/*****/

CLASS = 1

/*****/
/*      Name of File Type.                                           */
/*      -- Must be a string, max len 40 characters. Should          */
/*      be meaningful in that the name indicates what sort of      */
/*      data is stored within this file type.                        */
/*      -- There should be one of these for every File Type that    */
/*      can be associated with this ESDT.                             */
/*      Example                                                         */
/*      FILETYPE_NAME = "Instrument Band 7"                           */
/*****/

FILETYPE_NAME = ""

/*****/
/*      Maximum Number of Files under this Type.                     */
/*      -- Must be an integer.                                         */
/*      -- Indicates the maximum number of files for the            */
/*      specified File Type.                                          */
/*      -- Must be less than 1000.                                    */
/*      Example                                                         */
/*      MAXIMUM_NUM_FILES = 10                                         */
/*****/

MAXIMUM_NUM_FILES =

/*****/
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present          */
/*****/

END_OBJECT = FILETYPE

/*****/
/* Metadata Definition Object                                         */
/*                                                                    */
/* Metadata Defintion objects are required if there are to be Metadata */
/* Checks or Metadata Queries on this ESDT. The object defines the   */
/* metadata parameters and their types on which checks or queries will */
/* or can be performed.                                              */
/*                                                                    */
/* The actual values for the checks and/or queries are defined in the */
/* PGE ODL file. All that needs to be defined in this ESDT ODL file is */
/* the computer data type of the value. NOTE that there can be a      */
/* Metadata Definition object in the ESDT file and NO corresponding    */
/* Metadata Checks or Query object in the PGE ODL file. But if there  */

```

```

/* is a Metadata Checks or Query object in the PGE ODL file, there MUST */
/* be a corresponding Metadata Definition in the ESDT ODL file.          */
/*                                                                    */
/* This object is optional (only needed if there are Metadata Checks    */
/* or Metadata Query objects in the corresponding PGE ODL file).        */
/* There may be many of these objects per ESDT file.                    */
/*                                                                    */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present              */
/*****

OBJECT = METADATA_DEFINITION

/*****
/*                                                                    */
/*      Class (object counter, used only to distinguish objects)        */
/*      -- Must be an integer                                            */
/*      -- Must be unique in this file                                  */
/*      Example                                                         */
/*      CLASS = 1                                                         */
/*****

CLASS = 1

/*****
/*                                                                    */
/*      Parameter name for possible check or query                      */
/*      -- Must be a string, max len 40 characters                      */
/*      -- Must be identical to parm name read in PGE ODL file        */
/*      Example                                                         */
/*      PARM_NAME = "tbd_parm_name"                                     */
/*****

PARM_NAME = ""

/*****
/*                                                                    */
/*      Container name above the parameter to be checked/queried        */
/*      -- If not needed, should be set to "NONE".                      */
/*      -- Must be filled in (correctly) if there is a container        */
/*      object or a group surrounding the parameter specified          */
/*      by PARM_NAME. This is because Inspects on granules            */
/*      can only be performed at the highest level                     */
/*      object in the metadata tree.                                    */
/*      -- Must be a string, max len 100 characters                    */
/*      -- For Product Specific Attributes (PSAs) this must be        */
/*      set to "AdditionalAttributes"                                    */
/*      Example                                                         */
/*      For metadata that looks as follows:                             */
/*      GROUP = SOME_GROUP_NAME                                         */
/*      OBJECT = OBJECT_CONTAINER                                       */
/*      CLASS = "1"                                                     */
/*      OBJECT = PARAMETER_WE_ARE_QUERYING_ON                          */
/*      NUM_VAL = 1                                                      */
/*      VALUE = "Value we want"                                         */
/*      END_OBJECT = PARAMETER_WE_ARE_QUERYING_ON                      */
/*      END_OBJECT = OBJECT_CONTAINER                                   */
/*      END_GROUP = SOME_GROUP_NAME                                     */
/*      This parameter would be set as follows:                         */
/*      CONTAINER_NAME = "SOME_GROUP_NAME"

```

```

/*****/

CONTAINER_NAME = ""

/*****/
/*      Type of parameter for check or query      */
/*      -- Must be one of {FLOAT,INT,STR}          */
/*      Example                                     */
/*      TYPE = "INT"                               */
/*****/

TYPE = ""

/*****/
/* Optional Parameter. Defaults to empty string if not specified. */
/*                                     */
/*      Name of metadata parameter which provides a key into a    */
/*      a multi-containered object. Such an object is the         */
/*      MeasuredParameters group in the inventory metadata.       */
/*      -- Must be a string, max len 40 characters.              */
/*      -- Must be present in the ESDT ODL file for this ESDT.   */
/*      -- Is matched with KEY_PARAMETER_VALUE to determine      */
/*      the entry in a multi-containered metadata group. */
/*      -- For Product Specific Attributes (PSAs), this entry     */
/*      should NOT be specified. */
/*      -- Because an ESDT may be used by more than one PGE, it   */
/*      is possible to have more than one                          */
/*      KEY_PARAMETER_NAME/KEY_PARAMETER_VALUE pair               */
/*      (in multiple METADATA_DEFINITION objects)                 */
/*      per ESDT ODL File. Any PGE ODL file may only have        */
/*      a single KEY_PARAMETER_NAME/KEY_PARAMETER_VALUE pair. */
/*      -- For Metadata Checks or Queries within the              */
/*      MeasuredParameters group this should be set to the        */
/*      metadata field called "ParameterName". */
/*      Example: */
/*      KEY_PARAMETER_NAME = "ParameterName" */
/*****/

KEY_PARAMETER_NAME = ""

/*****/
/* Optional Parameter. Must be preset if KEY_PARAMETER_NAME exists. */
/* Defaults to the empty string if not specified. */
/*                                     */
/*      Value of metadata parameter which provides a key into a    */
/*      a multi-containered object. Such an object is the         */
/*      MeasuredParameters group in the inventory metadata.       */
/*      -- Must be a string, max len 80 characters.              */
/*      -- Must be present in the ESDT ODL file for this ESDT.   */
/*      -- Is matched with KEY_PARAMETER_NAME to determine      */
/*      the entry in a multi-containered metadata group. */
/*      -- For Product Specific Attributes (PSAs), this entry     */
/*      should NOT be specified. */
/*      -- Because an ESDT may be used by more than one PGE, it   */
/*      is possible to have more than one                          */
/*      KEY_PARAMETER_NAME/KEY_PARAMETER_VALUE pair               */
/*      (in multiple METADATA_DEFINITION objects)                 */
/*      per ESDT ODL File. Any PGE ODL file may only have        */

```

```

/*          a single KEY_PARAMETER_NAME/KEY_PARAMETER_VALUE pair. */
/*          -- For Metadata Checks or Queries within the          */
/*          MeasuredParameters group this should be set to the    */
/*          desired value of the metadata field called            */
/*          "ParameterName".                                     */
/*          Example:                                             */
/*          KEY_PARAMETER_VALUE = "LandCoverage"                  */
/*****/

KEY_PARAMETER_VALUE = ""

/*****/
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present      */
/*****/

END_OBJECT = METADATA_DEFINITION

/*****/
/* Metadata Definition object may be repeated as needed          */
/*****/

/*****/
/* Production Chain Object                                     */
/*          */
/* THIS OBJECT is only needed for those ESDTs that will be produced */
/* by On Demand Production Requests (Production Requests that are */
/* generated as a result of a request for an On Demand Product).    */
/*          */
/* The Production Chain object surrounds a list (in order) of the  */
/* PGEs needed to produce a granule of this ESDT. There may be one  */
/* PGE in the list (if that PGE takes in DYNAMIC External data and  */
/* produces this ESDT), or a chain of PGEs (if PGE A produces an  */
/* ESDT that is input to PGE B which produces THIS ESDT).          */
/*          */
/* The information contained within this object will only be used if */
/* an On Demand Request is for an ESDT which must have another     */
/* ESDT produced for the PGE that is to create the Product.        */
/*          */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present      */
/*****/

OBJECT = PRODUCTION_CHAIN

/*****/
/*          Class (object counter, used only to distinguish objects) */
/*          -- Must be an integer                                     */
/*          -- Must be unique in this file                           */
/*          Example                                                 */
/*          CLASS = 1                                               */
/*****/

CLASS = 1

/*****/
/* PGE In Chain Object                                     */
/*          */
/* THIS OBJECT defines a PGE that is part of the Production Chain */

```

```

/* used to produce this ESDT. */
/* */
/* The Production Chain object surrounds a list (in order) of the */
/* PGEs needed to produce a granule of this ESDT. There may be one */
/* PGE in the list (if that PGE takes in DYNAMIC External data and */
/* produces this ESDT), or a chain of PGEs (if PGE A produces an */
/* ESDT that is input to PGE B which produces THIS ESDT). */
/* */
/* The PGE_IN_CHAIN objects within the PRODUCTION_CHAIN object define */
/* the PGEs (in order) that need to be run to produce this ESDT. */
/* Only the PGE Name and Version are needed to identify the PGE, the */
/* Profile Id will be the one with the DEFEAUL_PROFILE flag set. */
/* */
/* The information contained within this object will only be used if */
/* an On Demand Request is for an ESDT which must have another */
/* ESDT produced for the PGE that is to create the Product. */
/* */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****

OBJECT = PGE_IN_CHAIN

/*****
/*
/*      Class (object counter) */
/*      -- Must be an integer */
/*      -- Must be unique in this file */
/*      -- Is used in this case to determine the order of the */
/*      PGEs. CLASS = 1 is the first PGE in the chain. */
/*      Example */
/*      CLASS = 1 */
/*****

CLASS = 1

/*****
/*
/*      PGE name */
/*      -- Must be a string, max len 10 characters */
/*      -- This is the name of the PGE that makes up one entry */
/*      in the chain of PGEs. */
/*      Example */
/*      PGE_NAME = "ssit" */
/*****

PGE_NAME = ""

/*****
/*
/*      PGE version */
/*      -- Must be a string, max len 5 characters */
/*      -- This is the version of the PGE that makes up one */
/*      entry in the chain of PGEs. */
/*      Example */
/*      PGE_VERSION = "1.0" */
/*****

PGE_VERSION = ""

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */

```

```

/*****/

END_OBJECT = PGE_IN_CHAIN

/*****/
/* Repeat PGE_IN_CHAIN objects as needed to make up the Production */
/* chain. */
/*****/

/*****/
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****/

END_OBJECT = PRODUCTION_CHAIN

/*****/
/* THE FOLLOWING LINE MUST NOT BE MODIFIED */
/*****/

END

```

A.1.3 ORBIT_ODL.template

```

/*****/
/* */
/*          TEMPLATE ORBIT MODEL METADATA ODL FILE */
/* */
/* */
/* The SSIT operator's responsibility is to copy this file over and */
/* edit it to add all necessary PDPS metadata values. */
/* */
/* All ORBIT MODEL ODL files must reside in directory */
/* $DPAT_RULE_SCIENCE_MD (set in the configuration files). */
/* */
/* The operator must add a value to the right of the "=" for each */
/* parameter. */
/* */
/* This file is only needed if the PGE has a period/boundary relating */
/* to orbit. */
/* */
/* There can be one or more ORBIT_MODEL objects defined in */
/* this file so that multiple orbits can be defined for the same */
/* platform. */
/* */
/* CHANGE LOG */
/* -- Added Orbit_Path_Number. 11/18/97 */
/* -- Changed acceptable Date Format. 01/05/98 */
/* -- Added another acceptable date format. 06/24/98 */
/* -- Updated length of PLATFORM. 08/13/98 */
/* -- Fixed where this file is located in above 10/01/98 */
/* comments. */
/*****/

/*****/
/*          Spacecraft platform name for the Orbit Model. */
/*          -- Must be a string, max len 25 characters */
/*          Example */

```

```

/*          PLATFORM = "TRMM"          */
/*****

PLATFORM = ""

/*****
/* Orbit Model object          */
/*          */
/* Defines the Orbit Model for a single orbit          */
/*          */
/* Replicate for the defining of multiple orbits for the same platform. */
/*          */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present          */
/*****

OBJECT = ORBIT_MODEL

/*****
/*          Class (object counter, used only to distinguish objects) */
/*          -- Must be an integer          */
/*          -- Must be unique in this file for this type of object */
/*          -- Must be greater than 0.          */
/*          Example          */
/*          CLASS = 1          */
/*****

CLASS = 1

/*****
/*          Number of the Orbit          */
/*          -- Must be an integer          */
/*          -- Must be >= 0          */
/*          Example          */
/*          ORBIT_NUMBER = 12          */
/*****

ORBIT_NUMBER =

/*****
/*          Path Number of the Orbit          */
/*          -- Must be an integer          */
/*          -- Must be >= 0 and <= 233          */
/*          Example          */
/*          ORBIT_PATH_NUMBER = 3          */
/*****

ORBIT_PATH_NUMBER =

/*****
/*          The period of the orbit (a duration).          */
/*          -- Must contain a single P=V string, where          */
/*          P is one of { MONTHS, WEEKS, DAYS, HOURS, MINS, SECS} */
/*          Example          */
/*          ORBIT_PERIOD = "HOURS=100"          */
/*****

ORBIT_PERIOD = " "

```



```

/*****
/*      The starting date/time of the orbit.                               */
/*      -- Must contain the date and time of the orbit.                   */
/*      -- The format for the date/time string can be one of the          */
/*      following:                                                         */
/*      "MMM DD YYYY HH:MM:SS", where                                     */
/*      YYYY=4 digit year, MMM=3 character abbreviation for               */
/*      Month, DD=2 digit Day, HH=Hours, MM=Minutes,                     */
/*      SS=Seconds. The time is accepted as UTC.                         */
/*      */
/*      "MM/DD/YYYY HH:MM:SS"                                             */
/*      YYYY=4 digit year, MM=2 digit Month,                             */
/*      DD=2 digit Day, HH=Hours, MM=Minutes,                             */
/*      SS=Seconds. The time is accepted as UTC.                         */
/*      -- NOTE that the format for the date of MM/DD/YY will            */
/*      no longer be accepted because it did not handle years            */
/*      after 1999 correctly.                                             */
/*      Example                                                            */
/*      ORBIT_START = "Oct 31 1996 22:01:55"                             */
*****/

ORBIT_START = " "

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present              */
*****/

END_OBJECT = ORBIT_MODEL

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED                               */
*****/

END

```

A.1.4 TILE_ODL.template

```

/*****
/*
/*      TEMPLATE TILE DEFINITION METADATA ODL FILE                       */
/*
/*
/* The SSIT operator's responsibility is to copy this file over and      */
/* edit it to add all necessary PDPS metadata values.                     */
/*
/* Each Tile Scheme used by a PGE must have a corresponding TILE         */
/* DEFINITION metadata ODL file.                                          */
/*
/* All TILE DEFINITION ODL files must reside in directory                */
/* $DPAT_RULE_SCIENCE_MD. Each must be named TILE_<tile scheme>.odl      */
/*
/* For a PGE to use a tile scheme, it must have SCHEDULE_TYPE =         */
/* "Tile". TILE_SCHEME_TYPE must equal the tiling scheme to be used.    */
/*
/* The operator must add a value to the right of the "=" for each        */
/* parameter.                                                              */

```

```

/* */
/* */
/* CHANGE LOG */
/* -- Removed the concept of CLUSTERS. 01/18/98 */
/* Added COORDINATE object. */
/* -- Updated various descriptions to make them better. 02/04/98 */
/* */
/* */
/*****/

/*****/
/* Tile Scheme */
/* -- Must be a string, max len 20 characters */
/* -- There can be NO spaces in the string. */
/* -- Tile Scheme must be identical to */
/* Tile Scheme used as part of ODL filename, */
/* which in turn was generated from the */
/* TILE_SCHEME_NAME in the PGE ODL file. */
/* Example */
/* TILE_SCHEME_NAME = "Earth_Squared" */
/*****/

TILE_SCHEME_NAME = " "

/*****/
/* Tile object */
/* */
/* Defines a tile for the scheme defined by TILE_SCHEME_NAME. */
/* Each tile must be defined seperately, with an ID, and associated */
/* coordinates. */
/* */
/* There should be a Tile object for every tile in the Tiling Scheme. */
/* */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****/

OBJECT = TILE

/*****/
/* Class (object counter, used only to distinguish objects) */
/* -- Must be an integer */
/* -- Must be unique in this file for this type of object */
/* -- Must be greater than 0. */
/* Example */
/* CLASS = 1 */
/*****/

CLASS = 1

/*****/
/* ID of Tile */
/* -- Must be an integer */
/* -- Must greater than 0 but less than max integer. */
/* -- Tiles should be listed sequentially (though no */
/* checking for this is done by software). */
/* -- This must be unique throughout the system. This */
/* means that if this Tile Id is defined in other Tile */
/* Schemes, it must have the same coordinates and */

```

```

/*          description.                                     */
/*          Example                                         */
/*          TILE_ID = 12                                    */
/*****

TILE_ID =

/*****
/*          Description of a Tile                           */
/*          -- A String of characters, max 255.             */
/*          -- Describes what the Tile is for, perhaps its */
/*          geographic location or area that it covers.     */
/*          Example                                         */
/*          TILE_DESCRIPTION = "Upper North America"       */
/*****

TILE_DESCRIPTION = ""

/*****
/* Tile Coordinate object                                  */
/*                                                         */
/* Defines a coordinate (Latitude and Longitude) for a tile. */
/*                                                         */
/* Each tile must have at least 4 TILE_COORDINATE objects defined. More */
/* (than 4) such objects are permitted to better define the tile. */
/*                                                         */
/* Coordidate objects must follow a clockwise sequence such that if */
/* lines were drawn between the points in the order they are given the */
/* desired shape would be drawn.                            */
/*                                                         */
/*                                                         */
/*                                                         */
/* For example:                                           */
/*          Coordinate 1                                Coordinate 2 */
/*          o----->o                                     */
/*          ^                                     | */
/*          |                                     | */
/*          |                                     v */
/*          o<-----o                                     */
/*          Coordidate 4                                Coordinate 3 */
/*                                                         */
/* Or:                                                    */
/*          Coordinate 2                                Coordinate 3 */
/*          o----->o                                     */
/*          ^                                     | */
/*          |                                     | */
/*          |                                     v */
/*          o<-----o                                     */
/*          Coordidate 1                                Coordinate 4 */
/*                                                         */
/*                                                         */
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****

```

```

OBJECT = TILE_COORDINATE

/*****
/*      Class (object counter, used only to distinguish objects)      */
/*      -- Must be an integer                                           */
/*      -- Must be unique in this file for this type of object        */
/*      -- Must be greater than 0.                                     */
/*      Example                                                         */
/*      CLASS = 1                                                       */
*****/

CLASS = 1

/*****
/*      Latitude Coordinate                                             */
/*      -- Must be one per Coordinate object.                          */
/*      -- Must be an float                                           */
/*      Example                                                         */
/*      LATITUDE = 12.15                                               */
*****/

LATITUDE =

/*****
/*      Longitude Coordinate                                           */
/*      -- Must be one per Coordinate object.                          */
/*      -- Must be an float                                           */
/*      Example                                                         */
/*      LONGITUDE = -43.22                                             */
*****/

LONGITUDE =

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present          */
*****/

END_OBJECT = TILE_COORDINATE

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present          */
*****/

END_OBJECT = TILE

/*****
/* THE FOLLOWING LINE MUST NOT BE MODIFIED                            */
*****/

END

```

A.1.5 PATHMAP_ODL.template

```

/*****
/*
/*      TEMPLATE PATHMAP DEFINITION METADATA ODL FILE
/*
/*
*****/

```

```

/* The SSIT operator's responsibility is to copy this file over and */
/* edit it to add all necessary PATH MAP metadata values.          */
/*                                                                    */
/* A PATHMAP defines the mapping between Absolute Path Number      */
/* a sequential numbering from 1-233 and Mapped Path Number which  */
/* is the interpreted 1-233 number.                                  */
/*                                                                    */
/* If a PGE defines a PATHMAP in the PGE ODL then there must be a  */
/* matching PATHMAP DEFINITION metadata ODL file and the PGE must have */
/* a SCHEDULE_TYPE = "Orbit".                                       */
/*                                                                    */
/* All PATHMAP DEFINITION ODL files must reside in directory       */
/* $DPAT_RULE_SCIENCE_MD. Each must be named                       */
/* PATHMAP_<Pathmap_Name>.odl. Note there can be NO spaces in the  */
/* Pathmap_Name because it is used as a filename.                  */
/*                                                                    */
/* For a PGE to use a PATHMAP, the PATHMAP_NAME parameter in the PGE */
/* ODL file must equal the Pathmap_Name to be used.                */
/*                                                                    */
/* The operator must add a value to the right of the "=" for each  */
/* parameter.                                                        */
/*                                                                    */
/* CHANGE LOG                                                        */
/*                                                                    */
/*****

/*****
/*      Spacecraft platform name for the Orbit Model.              */
/*      -- Must be a string, max len 20 characters                 */
/*      Example                                                      */
/*      PLATFORM = "TRMM"                                           */
/*****

PLATFORM = ""

/*****
/*      Pathmap Name                                                */
/*      -- Must be a string, max len 25 characters                 */
/*      -- There can be NO spaces in the string.                   */
/*      -- Pathmap Name must be identical to                       */
/*          Pathmap Name used as part of ODL filename,             */
/*          which in turn was generated from the                   */
/*          PATHMAP_NAME in the PGE ODL file.                       */
/*      Example                                                      */
/*      PATHMAP_NAME = "Some_Pathmap"                               */
/*****

PATHMAP_NAME = " "

/*****
/* Pathmap Entry Object                                             */
/*                                                                    */
/* Defines a mapping between Absolute Path Number                 */
/* a sequential numbering from 1-233 and Mapped Path Number which */
/* is the interpreted 1-233 number.                                  */
/*                                                                    */

```

```

/* There should be a Pathmap Entry object for each 1-233 Path Number. */
/* An error will be returned if one of the path numbers is missed. */
/*
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****

OBJECT = PATHMAP_ENTRY

/*****/
/*      Class (object counter, used only to distinguish objects) */
/*      -- Must be an integer */
/*      -- Must be unique in this file for this type of object */
/*      -- Must be greater than 0. */
/*      Example */
/*      CLASS = 1
/*****/

CLASS = 1

/*****/
/*      Absolute Path Number */
/*      -- Must be an integer */
/*      -- Must be between 1-233. */
/*      Example */
/*      ABSOLUTE_PATH = 20
/*****/

ABSOLUTE_PATH =

/*****/
/*      Mapped Path Number */
/*      -- Must be an integer. */
/*      -- Must be between 1-233. */
/*      Example */
/*      MAPPED_PATH = 27
/*****/

MAPPED_PATH = ""

/*****/
/* THE FOLLOWING LINE MUST NOT BE MODIFIED if it is present */
/*****/

END_OBJECT = PATHMAP_ENTRY

/*****/
/* THE FOLLOWING LINE MUST NOT BE MODIFIED */
/*****/
END

```

A.2 Typical ASTER PGE & ESDT ODL Files

Listings are provided for the following ASTER ODL files:

A.2.1 ASTER PGE ODL file for PGE_NAME BTS

A.2.2 ASTER ESDT ODL file for DATA_TYPE_NAME AST_LIB

A.2.3 ASTER ESDT ODL file for DATA_TYPE_NAME AST_ANC

A.2.4 ASTER ESDT ODL file for DATA_TYPE_NAME AST_04

A.2.5 ASTER ESDT ODL file for DATA_TYPE_NAME AST_09T

AST_LIB, AST_ANC and AST_04 are referenced within the PGE .

A typical ASTER PGE will differ from the example here by the PGE_NAME, the specific input/output files referenced, and runtime parameters. However, the overall structure of a given ASTER PGE ODL file would be similar to the one used here. (N.B. The ODL files shown here are associated with the ASTER version v2.2.34 software)

A.2.1 ASTER PGE BTS ODL

```
PGE_NAME = "BTS"
PGE_VERSION = "2.2h"
PROFILE_ID = 1
PROFILE_DESCRIPTION = "ASTER Brightness Temp with QA"
PLATFORM = "AM1"
INSTRUMENT = "ASTER"
MINIMUM_OUTPUTS = 1
SCHEDULE_TYPE = "Data"
PROCESSING_PERIOD = "SECS=1"
PROCESSING_BOUNDARY = "START_OF_SEC"
PGE_SSW_VERSION = "2.2h"

OBJECT = PCF_ENTRY
  CLASS = 11
  LOGICAL_ID = 15004
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "AST_LIB"
  DATA_TYPE_VERSION = "001"
  DATA_TYPE_REQUIREMENT = 1
  INPUT_TYPE = "Required"
  KEY_INPUT = "Y"
  NUMBER_NEEDED = 1
  /**** Entry needed for all I/O (except for Temporary) ****/
  /**** Only modify if multiple files and/or file types for this PCF entry ****/
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*Bright-Temp-LUT-v3.hdf*/
OBJECT = PCF_ENTRY
  CLASS = 12
  LOGICAL_ID = 15330
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "AST_ANC"
  DATA_TYPE_VERSION = "001"
  DATA_TYPE_REQUIREMENT = 1
  SCIENCE_GROUP = L1
  INPUT_TYPE = "Required"
```

```

    NUMBER_NEEDED = 1
/**** Entry needed for all I/O (except for Temporary) ****/
/**** Only modify if multiple files and/or file types for this PCF entry ****/
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
/**** "atmcorr-v3-dec.hdf" ****/
    CLASS = 29
    LOGICAL_ID = 15152
    PCF_FILE_TYPE = 3
    DATA_TYPE_NAME = "AST_ANC"
    DATA_TYPE_VERSION = "001"
    DATA_TYPE_REQUIREMENT = 1
    SCIENCE_GROUP = "L2"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
/**** Entry needed for all I/O (except for Temporary) ****/
/**** Only modify if multiple files and/or file types for this PCF entry ****/
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
/**** "FBA_Filter_File_1.cal" ****/
    CLASS = 30
    LOGICAL_ID = 15151
    PCF_FILE_TYPE = 3
    DATA_TYPE_NAME = "AST_ANC"
    DATA_TYPE_VERSION = "001"
    DATA_TYPE_REQUIREMENT = 1
    SCIENCE_GROUP = "O30"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
/**** Entry needed for all I/O (except for Temporary) ****/
/**** Only modify if multiple files and/or file types for this PCF entry ****/
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/* QA2-binning-intervals-v1.cal */
OBJECT = PCF_ENTRY
    CLASS = 13
    LOGICAL_ID = 15913
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "AST_ANC"
    DATA_TYPE_VERSION = "001"
    DATA_TYPE_REQUIREMENT = 1
    SCIENCE_GROUP = 098
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1

```



```

/**** Entry needed for all I/O (except for Temporary) ****/
/**** Only modify if multiple files and/or file types for this PCF entry ****/
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/**/
/* QA_thresholds.dat */
OBJECT = PCF_ENTRY
    CLASS = 14
    LOGICAL_ID = 15120
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "AST_ANC"
    DATA_TYPE_VERSION = "001"
    DATA_TYPE_REQUIREMENT = 1
    SCIENCE_GROUP = 097
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
/**** Entry needed for all I/O (except for Temporary) ****/
/**** Only modify if multiple files and/or file types for this PCF entry ****/
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*Output Product*/
OBJECT = PCF_ENTRY
    CLASS = 15
    LOGICAL_ID = 15010
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "AST_04"
    DATA_TYPE_VERSION = "001"
    YIELD = 1
    ASSOCIATED_MCF_ID = 15114
    SCIENCE_GROUP = "S1"
    INSTANCE = 0
    MINIMUM_SIZE = 5
    MAXIMUM_SIZE = 10
/**** Entry needed for all I/O (except for Temporary) ****/
/**** Only modify if multiple files and/or file types for this PCF entry ****/
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 126
    LOGICAL_ID = 15015
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "ASTALGRN"
    DATA_TYPE_VERSION = "001"
    YIELD = 1
    SCIENCE_GROUP = "S3"
    ASSOCIATED_MCF_ID = 15119

```

```

    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
/**** Entry needed for all I/O (except for Temporary) ****/
/**** Only modify if multiple files and/or file types for this PCF entry ****/
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 16
    LOGICAL_ID = 15602
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "PGE Major Version"
    PGE_PARAMETER_DEFAULT = "2"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 17
    LOGICAL_ID = 15603
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "PGE Minor Version"
    PGE_PARAMETER_DEFAULT = "2"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 18
    LOGICAL_ID = 16200
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "QA PGE Major Version"
    PGE_PARAMETER_DEFAULT = "2"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 19
    LOGICAL_ID = 16201
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "QA PGE Minor Version"
    PGE_PARAMETER_DEFAULT = "2"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 20
    LOGICAL_ID = 15604
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Brightness Temperature LUT"
    PGE_PARAMETER_DEFAULT = "3"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 139

```

```

    LOGICAL_ID = 15167
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "BrTtmp Lookup Table Version"
    PGE_PARAMETER_DEFAULT = "3"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 22
    LOGICAL_ID = 15165
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Atmos Corr. LUT Version"
    PGE_PARAMETER_DEFAULT = "3"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 24
    LOGICAL_ID = 15166
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "FBA Filters File Version"
    PGE_PARAMETER_DEFAULT = "3"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 21
    LOGICAL_ID = 15914
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "QA2 Binning Interval Version"
    PGE_PARAMETER_DEFAULT = "1"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 135
    LOGICAL_ID = 15320
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Alert File indirection"
    PGE_PARAMETER_DEFAULT = "15015:1"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 135
    LOGICAL_ID = 15321
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "PGE Name"
    PGE_PARAMETER_DEFAULT = "Brightness Temperature at the Sensor"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

END

```

A.2.2 ASTER ESDT AST_LIB ODL

```

DATA_TYPE_NAME = "AST_L1B"
DATA_TYPE_VERSION = "001"

```

```

INSTRUMENT = "ASTER"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "ASTER Level 1B Data Set Registered Radiance at the
                        Sensor"
PROVIDER = "EROS Data Center"
NOMINAL_SIZE = 120.0
PROCESSING_LEVEL = "L1"
HDF_DATA = "N"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "NONROUTINE"
SUPPLIER_NAME = "EDC"
PERIOD = "SECS=1"
DURATION = "SECS=1"
BOUNDARY = "START_OF_SEC"
DELAY = 1
SPATIAL_FLAG = "Y"
ARCHIVED_AT = "EDC"
PROCESSED_AT = "EDC"

END

```

A.2.3 ASTER ESDT AST Anc ODL

```

DATA_TYPE_NAME = "AST Anc"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "ASTER"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "HDF Ancillary data for ASTER"
PROVIDER = "Goddard Space Flight Center"
PROCESSING_LEVEL = "L1"
HDF_DATA = "Y"
NOMINAL_SIZE = 1.0
DYNAMIC_FLAG = "S"

END

```

A.2.4 ASTER ESDT AST_04 ODL

```

DATA_TYPE_NAME = "AST_04"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "ASTER"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "ASTER Level 2 Brightness Temperature at the Sensor"
PROVIDER = "Goddard Space Flight Center"
PROCESSING_LEVEL = "L1"
HDF_DATA = "Y"
NOMINAL_SIZE = 4.744895
DYNAMIC_FLAG = "I"
ARCHIVED_AT = "EDC"
PROCESSED_AT = "EDC"

END

```

A.2.5 ASTER ESDT AST_09T ODL

```
DATA_TYPE_NAME = "AST_09T"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "ASTER"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "ASTER Level 2 Surface Radiance Product (TIR)"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 9.439935
PROCESSING_LEVEL = "L1"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "EDC"
PERIOD = "SECS=1"
BOUNDARY = "START_OF_SEC"
DELAY = 1
SPATIAL_FLAG = "N"
ARCHIVED_AT = "EDC"
PROCESSED_AT = "EDC"

END
```

A.3 Typical MISR PGE & ESDT ODL Files

Listings are provided for the following MISR ODL files:

A.3.1 MISR PGE ODL file for PGE_NAME MPGE1 (M1AN)

A.3.2 MISR ESDT MISANCGM ODL

A.3.3 MISR ESDT MIRCCT ODL

A.3.4 MISR ESDT MISL0AN ODL

A.3.5 MISR ESDT ActSched ODL

A.3.6 MISR ESDT MIANCSSC ODL

A.3.7 MISR ESDT MIANCAGP ODL

A.3.8 MISR ESDT MIANPPAN ODL

A.3.9 MISR ESDT MISL0SY1 ODL

A.3.10 MISR ESDT MISL0SY2 ODL

A.3.11 MISR ESDT MISL0SY3 ODL

A.3.12 MISR ESDT MIRFOIAN ODL

A.3.13 MISR ESDT MIB2GEOP ODL

A.3.14 MISR ESDT MIANCARP ODL (Version# 001)

A.3.15 MISR ESDT MIANCARP ODL (Version# 002)

A.3.16 MISR ESDT MICNFG ODL

A.3.17 MISR ESDT AM1EPHN0 ODL

A.3.18 MISR ESDT AM1ATTNF ODL

A.3.19 MISR ESDT MIANRCCH ODL

A.3.20 MISR ESDT MIL1A ODL

A.3.21 MISR ESDT MISBR ODL

A.3.22 MISR ESDT MISQA ODL

A.3.23 MISR ESDT MI1B2T ODL

A.3.24 MISR ESDT MI1B2E ODL

A.3.25 MISR ESDT MIRCCM ODL

A.3.26 MISR ESDT MI1B1 ODL

A.3.27 MISR ESDT MIB1LM ODL

A typical MISR PGE will differ from the example here by the PGE_NAME, the specific input/output files referenced, and runtime parameters. However, the overall structure of a given MISR PGE ODL file would be similar to the one used here. (N.B. The ODL files shown here are associated with the MISR version v2.1.3 Patch 2 software)

A.3.1 MISR PGE MPGE1AN ODL (*profile #1*)

```
PGE_NAME = "M1AN"
PGE_VERSION = "21302"
PROFILE_ID = 1
PROFILE_DESCRIPTION = "MISR PGE 1 AN - Version V21302, SSI&T 17 March 2001"
PLATFORM = "AM1"
INSTRUMENT = "MISR"

/* MISR PGE 1 produces at a minimum 11 output files including QA          */
MINIMUM_OUTPUTS = 11

SCHEDULE_TYPE = "Orbit"
PROCESSING_PERIOD = "ORBITS=1"
PROCESSING_BOUNDARY = "START_OF_ORBIT"
PATHMAP_NAME = "MISR"

/* PGE_SSW_VERSION should match the PGE_VERSION                          */
PGE_SSW_VERSION = "21302"
```

```

OBJECT = EXIT_MESSAGE
  CLASS= 1
  EXIT_CODE = 0
  EXIT_MESSAGE = "CODE(0):  Successful Completion of MISR PGE 1 AN"
END_OBJECT = EXIT_MESSAGE
OBJECT = EXIT_MESSAGE
  CLASS= 2
  EXIT_CODE = 202
  EXIT_MESSAGE = "CODE(202):  Execution Failure of MISR PGE 1 AN"
END_OBJECT = EXIT_MESSAGE

/*****
/*          MISR PGE 1 AN  Inputs          */
/*  Inputs:                                */
/*  LID      ESDT.Version      Science Group      */
/*****
/*          MISR PGE 1 AN Inputs          */
/*  Inputs:                                */
/*  LID      ESDT.Version      Science Group      */
/*  190      MISANCGM.002      Dynamic External Input      */
/*  227      MIRCCT.001        L4003                  */
/*  243      MIRCCT.001        L9001                  */
/*  250      MICNFG.001        C1205                  */
/*  252      MICNFG.001        C1305                  */
/*  500      MISL0AN.001       Dynamic External Input      */
/*  555      MISL0SY1.001      Dynamic External Input      */
/*  556      MISL0SY2.001      Dynamic External Input      */
/*  557      MISL0SY3.001      Dynamic External Input      */
/*  599      MICNFG.001        C1415                  */
/*  1101     MICNFG.001        C1005                  */
/*  1120     ActSched.001      Dynamic External Input      */
/*  1301     MIANCSSC.001      C0002                  */
/*  1304     MIANCAGP.001      L0002 - Path Dependent Static File      */
/*  1305     MIANPPAN.001      L1001                  */
/*  1306     MIRFOIAN.001      L1001                  */
/*  1334     MIB2GEOP.001      Dynamic Internal Input      */
/*  1500     MIANCARP.001      C0010                  */
/*  1501     MIANCARP.001      C0011                  */
/*  1502     MIANCARP.002      Dynamic External Input      */
/*  1503     MIANCARP.001      C0012                  */
/*  1984     MICNFG.001        C1105                  */
/*  10501    AM1EPHN0.001      Dynamic Internal Input      */
/*  10502    AM1ATTNF.001      Dynamic Internal Input      */
/*
/*****

/*          PCF Entry for 190:MISANCGM          */
/*  MISR Ancillary Dataset for Camera Model  OBJECT = PCF_ENTRY      */
/*  CLASS = 11
/*  LOGICAL_ID = 190
/*  PCF_FILE_TYPE = 1
/*  DATA_TYPE_NAME = "MISANCGM"
/*  DATA_TYPE_VERSION = "002"
/*  MIN_GRANULES_REQUIRED = 1
/*  MAX_GRANULES_REQUIRED = 1

```

```

INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
CLOSEST_QUERY_OFFSET = "WEEKS=9"
CLOSEST_QUERY_DIRECTION = "Backward"
CLOSEST_QUERY_RETRIES = 6
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*      PCF Entry for 227:MIRCCT      */
/*      MISR RC Thresholds datasetOBJECT = PCF_ENTRY      */
CLASS = 12
LOGICAL_ID = 227
PCF_FILE_TYPE = 1
DATA_TYPE_NAME = "MIRCCT"
DATA_TYPE_VERSION = "001"
MIN_GRANULES_REQUIRED = 1
MAX_GRANULES_REQUIRED = 1
SCIENCE_GROUP = "L4003"
INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*      PCF Entry for 243:MIRCCT      */
/*      MISR RC Thresholds datasetOBJECT = PCF_ENTRY      */
CLASS = 13
LOGICAL_ID = 243
PCF_FILE_TYPE = 1
DATA_TYPE_NAME = "MIRCCT"
DATA_TYPE_VERSION = "001"
MIN_GRANULES_REQUIRED = 1
MAX_GRANULES_REQUIRED = 1
SCIENCE_GROUP = "L9001"
INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```



```

/*      PCF Entry for 500:MISL0AN      */
/*      L0 AN data                      */
OBJECT = PCF_ENTRY
  CLASS = 14
  LOGICAL_ID = 500
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "MISL0AN"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 2
  INPUT_TYPE = "Required"
/*  ALIGN DPR TIME WITH INPUT TIME = "Y" */
ALIGN DPR TIME WITH INPUT TIME = "N"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
/* 4PY version */
OBJECT = FILETYPE
  FILETYPE_NAME = "Multi-File Granule"
  CLASS = 1
END_OBJECT = FILETYPE
OBJECT = AUXILIARY_LOGICAL_ID
  CLASS = 1
  AUX_LOGICAL_ID = 501
END_OBJECT = AUXILIARY_LOGICAL_ID
END_OBJECT = PCF_ENTRY

/*      PCF Entry for 1120:ActSched      */
/*      Detailed Activity Schedule from EMOS      */
OBJECT = PCF_ENTRY
  CLASS = 16
  LOGICAL_ID = 1120
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "ActSched"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 2
  INPUT_TYPE = "Required"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*      PCF Entry for 1301:MIANCSSC      */
/*      MISR CSSC dataset                  */
OBJECT = PCF_ENTRY
  CLASS = 17
  LOGICAL_ID = 1301
  PCF_FILE_TYPE = 1

```

```

DATA_TYPE_NAME = "MIANCSSC"
DATA_TYPE_VERSION = "001"
MIN_GRANULES_REQUIRED = 1
MAX_GRANULES_REQUIRED = 1
SCIENCE_GROUP = "C0002"
INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1304:MIANCAGP      */
/*      MISR Ancillary Geographic Product */
OBJECT = PCF_ENTRY
    CLASS = 18
    LOGICAL_ID = 1304
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MIANCAGP"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "L0002"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Metadata"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
    OBJECT = METADATA_QUERY
        CLASS = 1
        PARM_NAME = "SP_AM_PATH_NO"
        OPERATOR = "=="
        VALUE = "999"
        DATABASE_QUERY = "PATH NUMBER"
    END_OBJECT = METADATA_QUERY
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1305:MIANPPAN      */
/*      MISR Project Parameters (PP) dataset */
OBJECT = PCF_ENTRY
    CLASS = 19
    LOGICAL_ID = 1305
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MIANPPAN"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1

```

```

MAX_GRANULES_REQUIRED = 1
SCIENCE_GROUP = "L1001"
INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Metadata"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
OBJECT = METADATA_QUERY
    CLASS = 1
    PARM_NAME = "SP_AM_PATH_NO"
    OPERATOR = "=="
    VALUE = "999"
    DATABASE_QUERY = "PATH NUMBER"
END_OBJECT = METADATA_QUERY
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 555:MISL0SY1      */
/*      L0 Out of Synch data            */
OBJECT = PCF_ENTRY
    CLASS = 20
    LOGICAL_ID = 555
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MISL0SY1"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 2
    INPUT_TYPE = "Optional"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Multi-File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
    OBJECT = AUXILIARY_LOGICAL_ID
        CLASS = 1
        AUX_LOGICAL_ID = 5551
    END_OBJECT = AUXILIARY_LOGICAL_ID
    OBJECT = OPTIONAL_INPUT
        CLASS = 1
        CATEGORY = "Out of Sync SY1"
        ORDER = 1
        RUNTIME_PARM_ID = 555
        TIMER = "SECS=10"
        TEMPORAL = "N"
    END_OBJECT = OPTIONAL_INPUT
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 556:MISL0SY2      */

```

```

/*          L0 Out of Synch data                                */
OBJECT = PCF_ENTRY
  CLASS = 21
  LOGICAL_ID = 556
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "MISL0SY2"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 2
  INPUT_TYPE = "Optional"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"
  OBJECT = FILETYPE
    FILETYPE_NAME = "Multi-File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
  OBJECT = AUXILIARY_LOGICAL_ID
    CLASS = 1
    AUX_LOGICAL_ID = 5561
  END_OBJECT = AUXILIARY_LOGICAL_ID
  OBJECT = OPTIONAL_INPUT
    CLASS = 1
    CATEGORY = "Out of Sync SY2"
    ORDER = 1
    RUNTIME_PARM_ID = 556
    TIMER = "SECS=10"
    TEMPORAL = "N"
  END_OBJECT = OPTIONAL_INPUT
END_OBJECT = PCF_ENTRY

```

```

/*          PCF Entry for 557:MISL0SY3                          */
/*          L0 Out of Synch data                                */
OBJECT = PCF_ENTRY
  CLASS = 22
  LOGICAL_ID = 557
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "MISL0SY3"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 2
  INPUT_TYPE = "Optional"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"
  OBJECT = FILETYPE
    FILETYPE_NAME = "Multi-File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
  OBJECT = AUXILIARY_LOGICAL_ID
    CLASS = 1
    AUX_LOGICAL_ID = 5571
  END_OBJECT = AUXILIARY_LOGICAL_ID
  OBJECT = OPTIONAL_INPUT

```

```

        CLASS = 1
        CATEGORY = "Out of Sync SY3"
        ORDER = 1
        RUNTIME_PARM_ID = 557
        TIMER = "SECS=10"
        TEMPORAL = "N"
    END_OBJECT = OPTIONAL_INPUT
END_OBJECT = PCF_ENTRY

```

```

/*          PCF Entry for 1306:MIRFOIAN          */
/*          MISR Reference Orbit Imagery          */
OBJECT = PCF_ENTRY
    CLASS = 110
    LOGICAL_ID = 1306
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MIRFOIAN"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "L1001"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Metadata"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
    OBJECT = METADATA_QUERY
        CLASS = 1
        PARM_NAME = "SP_AM_PATH_NO"
        OPERATOR = "=="
        VALUE = "999"
        DATABASE_QUERY = "PATH NUMBER"
    END_OBJECT = METADATA_QUERY
END_OBJECT = PCF_ENTRY

```

```

/*          PCF Entry for 1334:MIB2GEOP          */
/*          MISR Geometric Parameters          */
OBJECT = PCF_ENTRY
    CLASS = 111
    LOGICAL_ID = 1334
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MIB2GEOP"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE

```

```

        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1500:MIANCARP      */
/*      MISR Ancillary Radiometric Product      */
OBJECT = PCF_ENTRY
    CLASS = 112
    LOGICAL_ID = 1500
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MIANCARP"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "C0010"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1501:MIANCARP      */
/*      MISR Ancillary Radiometric Product      */
OBJECT = PCF_ENTRY
    CLASS = 113
    LOGICAL_ID = 1501
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MIANCARP"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "C0011"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1502:MIANCARP      */
/*      MISR Ancillary Radiometric Product      */
OBJECT = PCF_ENTRY

```

```

CLASS = 114
LOGICAL_ID = 1502
PCF_FILE_TYPE = 1
DATA_TYPE_NAME = "MIANCARP"
DATA_TYPE_VERSION = "002"
MIN_GRANULES_REQUIRED = 1
MAX_GRANULES_REQUIRED = 1
INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
CLOSEST_QUERY_OFFSET = "WEEKS=8"
CLOSEST_QUERY_DIRECTION = "Backward"
CLOSEST_QUERY_RETRIES = 10
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1503:MIANCARP      */
/*      MISR Ancillary Radiometric Product      */
OBJECT = PCF_ENTRY
    CLASS = 115
    LOGICAL_ID = 1503
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MIANCARP"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "C0012"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 250:MICNFG      */
/*      MISR RCCM configuration file      */
OBJECT = PCF_ENTRY
    CLASS = 116
    LOGICAL_ID = 250
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MICNFG"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "C1205"

```

```

INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 252:MICNFG      */
/*      MISR GRP configuration file    */
OBJECT = PCF_ENTRY
    CLASS = 117
    LOGICAL_ID = 252
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MICNFG"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "C1305"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 599:MICNFG      */
/*      MISR RAP configuration file    */
OBJECT = PCF_ENTRY
    CLASS = 118
    LOGICAL_ID = 599
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MICNFG"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "C1415"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```



```

/*      PCF Entry for 1984:MICNFG      */
/*      MISR RP configuration file      */
OBJECT = PCF_ENTRY
  CLASS = 119
  LOGICAL_ID = 1984
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "MICNFG"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 1
  SCIENCE_GROUP = "C1105"
  INPUT_TYPE = "Required"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/**** Attitude/Ephemeris/DEM entry.  Please delete if not used by PGE. **/
/*      PCF Entry for 10501:AM1EPHN0      */
/*      Ephemeris data generated from DPREP      */
/*      External Data Source      */
OBJECT = PCF_ENTRY
  CLASS = 120
  LOGICAL_ID = 10501
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "AM1EPHN0"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 2
  INPUT_TYPE = "Required"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/**** Attitude/Ephemeris/DEM entry.  Please delete if not used by PGE. **/
/*      PCF Entry for 10502:AM1ATTNF      */
/*      Attitude data generated by DPREP      */
/*      External Data Source      */
OBJECT = PCF_ENTRY
  CLASS = 121

```

```

LOGICAL_ID = 10502
PCF_FILE_TYPE = 1
DATA_TYPE_NAME = "AM1ATTNF"
DATA_TYPE_VERSION = "001"
MIN_GRANULES_REQUIRED = 1
MAX_GRANULES_REQUIRED = 2
INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1101:MICNFG      */
/*      MISR PCS configuration file     */
OBJECT = PCF_ENTRY
    CLASS = 132
    LOGICAL_ID = 1101
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MICNFG"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "C1005"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*****
/*  MISR PGE 1 AN Outputs      */
/*      Output:                */
/*      LID      ESDT.Version   Science Group   Associated MCF      */
/*      251      MIANRCCH.002   S4              1136                */
/*      600      MIL1A.001      S5              1130                */
/*      610      MISBR.002      S6              1138                */
/*      611      MISBR.002      S7              1138                */
/*      650      MISQA.002      S10             1137                */
/*      1375     MI1B2T.001     S2              1133                */
/*      1376     MI1B2E.001     S1              1134                */
/*      1377     MIRCCM.001     S3              1135                */
/*      1335     MISQA.002      S11             11371               */
/*      1336     MISQA.002      S12             11372               */
/*      1337     MISQA.002      S13             11373               */
/*      1976     MI1B1.001     S8              1140                */
/*      1983     MIB1LM.001     S9              1131                */

```

```

/*      1985      MISQA.002      S14      11374      */
/*      1986      MISQA.002      S15      11375      */
/*      */
/*****

/*      PCF Entry for 251:MIANRCCH      */
/*      MISR RC Histogram file      */
OBJECT = PCF_ENTRY
  CLASS = 136
  LOGICAL_ID = 251
  PCF_FILE_TYPE = 2
  DATA_TYPE_NAME = "MIANRCCH"
  DATA_TYPE_VERSION = "002"
  MIN_GRANULE_YIELD = 1
  MAX_GRANULE_YIELD = 1
  ASSOCIATED_MCF_ID = 1136
  SCIENCE_GROUP = "S4"
  INSTANCE = 0
  MINIMUM_SIZE = 0
  MAXIMUM_SIZE = 0
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*      PCF Entry for 600:MIL1A      */
/*      MISR L1A Product      */
OBJECT = PCF_ENTRY
  CLASS = 137
  LOGICAL_ID = 600
  PCF_FILE_TYPE = 2
  DATA_TYPE_NAME = "MIL1A"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULE_YIELD = 1
  MAX_GRANULE_YIELD = 1
  ASSOCIATED_MCF_ID = 1130
  SCIENCE_GROUP = "S5"
  INSTANCE = 0
  MINIMUM_SIZE = 0
  MAXIMUM_SIZE = 0
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*      PCF Entry for 610:MISBR      */
/*      MISR Browse data HDF file      */
OBJECT = PCF_ENTRY
  CLASS = 138
  LOGICAL_ID = 610
  PCF_FILE_TYPE = 2

```

```

DATA_TYPE_NAME = "MISBR"
DATA_TYPE_VERSION = "002"
MIN_GRANULE_YIELD = 0
MAX_GRANULE_YIELD = 1
ASSOCIATED_MCF_ID = 1138
SCIENCE_GROUP = "S6"
INSTANCE = 0
MINIMUM_SIZE = 0
MAXIMUM_SIZE = 0
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 611:MISBR      */
/*      MISR Browse data JPEG file  */
OBJECT = PCF_ENTRY
    CLASS = 139
    LOGICAL_ID = 611
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MISBR"
    DATA_TYPE_VERSION = "002"
    MIN_GRANULE_YIELD = 0
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 1138
    SCIENCE_GROUP = "S7"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 650:MISQA      */
/*      MISR L1A QA                  */
OBJECT = PCF_ENTRY
    CLASS = 140
    LOGICAL_ID = 650
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MISQA"
    DATA_TYPE_VERSION = "002"
    MIN_GRANULE_YIELD = 0
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 1137
    SCIENCE_GROUP = "S10"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
```

```
/*      PCF Entry for 1375:MI1B2T      */
/*      MISR L1B2 Terrain data      */
OBJECT = PCF_ENTRY
CLASS = 141
LOGICAL_ID = 1375
PCF_FILE_TYPE = 2
DATA_TYPE_NAME = "MI1B2T"
DATA_TYPE_VERSION = "001"
MIN_GRANULE_YIELD = 1
MAX_GRANULE_YIELD = 1
ASSOCIATED_MCF_ID = 1133
SCIENCE_GROUP = "S2"
INSTANCE = 0
MINIMUM_SIZE = 0
MAXIMUM_SIZE = 0
DISTINCT_VALUE = "AN"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
```

```
/*      PCF Entry for 1376:MI1B2E      */
/*      MISR L1B2 Ellipsoid data      */
OBJECT = PCF_ENTRY
CLASS = 142
LOGICAL_ID = 1376
PCF_FILE_TYPE = 2
DATA_TYPE_NAME = "MI1B2E"
DATA_TYPE_VERSION = "001"
MIN_GRANULE_YIELD = 1
MAX_GRANULE_YIELD = 1
ASSOCIATED_MCF_ID = 1134
SCIENCE_GROUP = "S1"
INSTANCE = 0
MINIMUM_SIZE = 0
MAXIMUM_SIZE = 0
DISTINCT_VALUE = "AN"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
```

```
/*      PCF Entry for 1377:MIRCCM      */
/*      MISR L1B2 RCCM data      */
OBJECT = PCF_ENTRY
CLASS = 143
LOGICAL_ID = 1377
```

```

PCF_FILE_TYPE = 2
DATA_TYPE_NAME = "MIRCCM"
DATA_TYPE_VERSION = "001"
MIN_GRANULE_YIELD = 1
MAX_GRANULE_YIELD = 1
ASSOCIATED_MCF_ID = 1135
SCIENCE_GROUP = "S3"
INSTANCE = 0
MINIMUM_SIZE = 0
MAXIMUM_SIZE = 0
DISTINCT_VALUE = "AN"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1335:MISQA      */
/*      MISR L1B2 Terrain QA data     */
OBJECT = PCF_ENTRY
    CLASS = 144
    LOGICAL_ID = 1335
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MISQA"
    DATA_TYPE_VERSION = "002"
    MIN_GRANULE_YIELD = 0
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 11371
    SCIENCE_GROUP = "S11"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1336:MISQA      */
/*      MISR L1B2 Ellipsoid QA data    */
OBJECT = PCF_ENTRY
    CLASS = 145
    LOGICAL_ID = 1336
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MISQA"
    DATA_TYPE_VERSION = "002"
    MIN_GRANULE_YIELD = 0
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 11372
    SCIENCE_GROUP = "S12"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"

```

```

        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1337:MISQA      */
/*      MISR L1B2 RCCM QA data      */
OBJECT = PCF_ENTRY
    CLASS = 146
    LOGICAL_ID = 1337
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MISQA"
    DATA_TYPE_VERSION = "002"
    MIN_GRANULE_YIELD = 0
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 11373
    SCIENCE_GROUP = "S13"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1976:MI1B1      */
/*      MISR L1B1 Radiometric Product */
OBJECT = PCF_ENTRY
    CLASS = 147
    LOGICAL_ID = 1976
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MI1B1"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULE_YIELD = 1
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 1140
    SCIENCE_GROUP = "S8"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1983:MIB1LM      */
/*      MISR L1B1 Local Mode data      */
OBJECT = PCF_ENTRY
    CLASS = 148
    LOGICAL_ID = 1983
    PCF_FILE_TYPE = 2

```

```

DATA_TYPE_NAME = "MIB1LM"
DATA_TYPE_VERSION = "001"
MIN_GRANULE_YIELD = 0
MAX_GRANULE_YIELD = 6
ASSOCIATED_MCF_ID = 1131
SCIENCE_GROUP = "S9"
INSTANCE = 0
MINIMUM_SIZE = 0
MAXIMUM_SIZE = 0
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1985:MISQA      */
/*      MISR L1B1 QA data             */
OBJECT = PCF_ENTRY
    CLASS = 149
    LOGICAL_ID = 1985
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MISQA"
    DATA_TYPE_VERSION = "002"
    MIN_GRANULE_YIELD = 0
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 11374
    SCIENCE_GROUP = "S14"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

/*      PCF Entry for 1986:MISQA      */
/*      MISR L1B1 Local Mode QA       */
OBJECT = PCF_ENTRY
    CLASS = 150
    LOGICAL_ID = 1986
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MISQA"
    DATA_TYPE_VERSION = "002"
    MIN_GRANULE_YIELD = 0
    MAX_GRANULE_YIELD = 6
    ASSOCIATED_MCF_ID = 11375
    SCIENCE_GROUP = "S15"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```



```
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
```

```
OBJECT = PCF_ENTRY
CLASS = 151
LOGICAL_ID = 292
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Product version"
PGE_PARAMETER_DEFAULT = "0007"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
```

```
OBJECT = PCF_ENTRY
CLASS = 152
LOGICAL_ID = 295
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Camera"
PGE_PARAMETER_DEFAULT = "An"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
```

```
OBJECT = PCF_ENTRY
CLASS = 153
LOGICAL_ID = 620
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Ascii met file for HDF browse"
PGE_PARAMETER_DEFAULT = "610:1"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
```

```
OBJECT = PCF_ENTRY
CLASS = 154
LOGICAL_ID = 621
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Ascii met file for JPEG browse"
PGE_PARAMETER_DEFAULT = "611:1"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
```

```
OBJECT = PCF_ENTRY
CLASS = 155
LOGICAL_ID = 1102
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "which pge"
PGE_PARAMETER_DEFAULT = "MISR_PGE01"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
```

```
OBJECT = PCF_ENTRY
CLASS = 156
LOGICAL_ID = 1104
PCF_FILE_TYPE = 5
```

```

    PGE_PARAMETER_NAME = "Orbit number"
    PGE_PARAMETER_DEFAULT = "999999"
    PGE_PARAMETER_DYNAMIC_VALUE = "ORBIT NUMBER"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 157
    LOGICAL_ID = 1103
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Orbit path"
    PGE_PARAMETER_DEFAULT = "999"
    PGE_PARAMETER_DYNAMIC_VALUE = "PATH NUMBER"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 158
    LOGICAL_ID = 10119
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Disabled status code list"
    PGE_PARAMETER_DEFAULT = "35870 163843126 163843127 163842611 163842612
166300169 164662287"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

END

```

A.3.2 MISR ESDT MISANCGM ODL

```

DATA_TYPE_NAME = "MISANCGM"
DATA_TYPE_VERSION = "002"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "Camera Geometric Model for Level 1B2"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 1.0
PROCESSING_LEVEL = "L1"
HDF_DATA = "N"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "NONROUTINE"
SUPPLIER_NAME = "LARC"
/* BOUNDARY = "START_OF_YEAR" */
/* PERIOD = "YEARS=5" */
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END

```

A.3.3 MISR ESDT MIRCCT ODL

```
DATA_TYPE_NAME = "MIRCCT"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Radiometric Camera-by-Camera Threshold dataset"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 10.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "Y"
DYNAMIC_FLAG = "S"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"

END
```

A.3.4 MISR ESDT MISLOAN ODL

```
DATA_TYPE_NAME = "MISLOAN"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Level 0 CCD Science Data AN Camera"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 1000.0
PROCESSING_LEVEL = "L0"
HDF_DATA = "N"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
/* PERIOD = "ORBITS=1" */
PERIOD = "HOURS=2"
/* BOUNDARY = "START_OF_ORBIT" */
BOUNDARY = "START_OF_DAY"
DURATION = "HOURS=2"
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"
OBJECT = FILETYPE
    CLASS = 2
    FILETYPE_NAME = "Multi-File Granule"
    MAXIMUM_NUM_FILES = 2
END_OBJECT = FILETYPE

END
```

A.3.5 MISR ESDT ActSched ODL

```
DATA_TYPE_NAME = "ActSched"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "Detailed Activity Schedule"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 1.0
/* Changed by Jim Galasso 10/9/1999 */
/* Processing Level cannot be L0 multiple files using the same LID */
/* Change of Processing level is to support PGE processing when 2 DAS */
/* files are required because the PGE's DPR times span 2 files */
PROCESSING_LEVEL = "SCHED"
HDF_DATA = "N"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
/* Q: Should the supplier of the DAS be identified as EMOS? */
SUPPLIER_NAME = "EMOS"
PERIOD = "DAYS=1"
/* Boundary set for DAS files to be 2000 to 2000 each day */
BOUNDARY = "START_OF_DAY-14400"
DURATION = "HOURS=24"
DELAY = 3600
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END
```

A.3.6 MISR ESDT MIANCSSC ODL

```
DATA_TYPE_NAME = "MIANCSSC"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Cloud Screening Surface Classification dataset"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 5.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "Y"
DYNAMIC_FLAG = "S"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END
```

A.3.7 MISR ESDT MIANCAGP ODL

```
DATA_TYPE_NAME = "MIANCAGP"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Ancillary Geographic Product"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 110.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "Y"
DYNAMIC_FLAG = "S"
SPATIAL_FLAG = "Y"
OBJECT = METADATA_DEFINITION
    CLASS = 1
    PARM_NAME = "SP_AM_PATH_NO"
    CONTAINER_NAME = "AdditionalAttributes"
    TYPE = "STR"
END_OBJECT = METADATA_DEFINITION
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT

END
```

A.3.8 MISR ESDT MIANPPAN ODL

```
DATA_TYPE_NAME = "MIANPPAN"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Projection Parameters Ancillary Dataset, Camera AN"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 310.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "Y"
DYNAMIC_FLAG = "S"
SPATIAL_FLAG = "Y"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
OBJECT = METADATA_DEFINITION
    CLASS = 1
    PARM_NAME = "SP_AM_PATH_NO"
    CONTAINER_NAME = "AdditionalAttributes"
    TYPE = "STR"
END_OBJECT = METADATA_DEFINITION

END
```

A.3.9 MISR ESDT MISL0SY1 ODL

```
DATA_TYPE_NAME = "MISL0SY1"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Out of Sync L0 CCSDS packets for APID = 373"
PROVIDER = "Langley Research Center"
PROCESSING_LEVEL = "L0"
HDF_DATA = "N"
/* Q: NOMINAL_SIZE ???? */
NOMINAL_SIZE = 5.9
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
PERIOD = "ORBITS=1"
/* PERIOD = "HOURS=2" */
BOUNDARY = "START_OF_ORBIT"
/* BOUNDARY = "START_OF_DAY+3600" */
DURATION = "ORBITS=1"
/* DURATION = "HOURS=2" */
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"
OBJECT = FILETYPE
    CLASS = 2
    FILETYPE_NAME = "Multi-File Granule"
    MAXIMUM_NUM_FILES = 2
END_OBJECT = FILETYPE

END
```

A.3.10 MISR ESDT MISL0SY2 ODL

```
DATA_TYPE_NAME = "MISL0SY2"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Out of Sync L0 CCSDS packets for APID = 374"
PROVIDER = "Langley Research Center"
PROCESSING_LEVEL = "L0"
HDF_DATA = "N"
/* Q: NOMINAL_SIZE ???? */
NOMINAL_SIZE = 5.9
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
PERIOD = "ORBITS=1"
/* PERIOD = "HOURS=2" */
BOUNDARY = "START_OF_ORBIT"
/* BOUNDARY = "START_OF_DAY+3600" */
DURATION = "ORBITS=1"
```

```

/* DURATION = "HOURS=2" */
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"
OBJECT = FILETYPE
    CLASS = 2
    FILETYPE_NAME = "Multi-File Granule"
    MAXIMUM_NUM_FILES = 2
END_OBJECT = FILETYPE

END

```

A.3.11 MISR ESDT MISL0SY3 ODL

```

DATA_TYPE_NAME = "MISL0SY3"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Out of Sync L0 CCSDS packets for APID = 378"
PROVIDER = "Langley Research Center"
PROCESSING_LEVEL = "L0"
HDF_DATA = "N"
/* Q: NOMINAL_SIZE ???? */
NOMINAL_SIZE = 5.9
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
PERIOD = "ORBITS=1"
/* PERIOD = "HOURS=2" */
BOUNDARY = "START_OF_ORBIT"
/* BOUNDARY = "START_OF_DAY+3600" */
DURATION = "ORBITS=1"
/* DURATION = "HOURS=2" */
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"
OBJECT = FILETYPE
    CLASS = 2
    FILETYPE_NAME = "Multi-File Granule"
    MAXIMUM_NUM_FILES = 2
END_OBJECT = FILETYPE

END

```

A.3.12 MISR ESDT MIRFOIAN ODL

```
DATA_TYPE_NAME = "MIRFOIAN"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Reference Orbit Imagery Ancillary Dataset,
Camera AN"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 280.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "Y"
DYNAMIC_FLAG = "S"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
OBJECT = METADATA_DEFINITION
    CLASS = 1
    PARM_NAME = "SP_AM_PATH_NO"
    CONTAINER_NAME = "AdditionalAttributes"
    TYPE = "STR"
END_OBJECT = METADATA_DEFINITION
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END
```

A.3.13 MISR ESDT MIB2GEOP ODL

```
DATA_TYPE_NAME = "MIB2GEOP"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Geometric Parameters"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 6.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
SPATIAL_FLAG = "Y"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END
```

A.3.14 MISR ESDT MIANCARP ODL (Version# 001)

```
DATA_TYPE_NAME = "MIANCARP"
DATA_TYPE_VERSION = "001"
```



```

INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Ancillary Radiometric Product (ARP)"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 5.0
PROCESSING_LEVEL = "All"
HDF_DATA = "Y"
DYNAMIC_FLAG = "S"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT

END

```

A.3.15 MISR ESDT MIANCARP ODL (Version# 002)

```

DATA_TYPE_NAME = "MIANCARP"
DATA_TYPE_VERSION = "002"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Ancillary Radiometric Product (ARP)"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 5.0
PROCESSING_LEVEL = "All"
HDF_DATA = "Y"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "NONROUTINE"
/* PERIOD = "MONTHS=2" */
/* BOUNDARY = "START_OF_MONTH" */
SUPPLIER_NAME = "LARC"
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END

```

A.3.16 MISR ESDT MICNFG ODL

```

DATA_TYPE_NAME = "MICNFG"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Configuration File for all PGEs"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 0.5
PROCESSING_LEVEL = "All"
HDF_DATA = "N"
DYNAMIC_FLAG = "S"

```

```

SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT

END

```

A.3.17 MISR ESDT AM1EPHN0 ODL

```

DATA_TYPE_NAME = "AM1EPHN0"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "All"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "AM-1 L0/FDD Ephemeris data in Toolkit format"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 2.0
PROCESSING_LEVEL = "L1"
DYNAMIC_FLAG = "I"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
HDF_DATA = "N"

END

```

A.3.18 MISR ESDT AM1ATTNF ODL

```

DATA_TYPE_NAME = "AM1ATTNF"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "All"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "AM-1 FDD Attitude data in Toolkit format"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 2.0
PROCESSING_LEVEL = "L1"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
HDF_DATA = "N"

END

```

A.3.19 MISR ESDT MIANRCCH ODL

```
DATA_TYPE_NAME = "MIANRCCH"
DATA_TYPE_VERSION = "002"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Radiometric Camera-by-Camera Histogram Dataset"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 3.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "N"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END
```

A.3.20 MISR ESDT MIL1A ODL

```
DATA_TYPE_NAME = "MIL1A"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Level 1A CCD Science data, all cameras"
PROVIDER = "Langley Research Center"
/*      Q: Need to find the correct nominal file size for MIL1A      */
/* NOMINAL_SIZE = 498.0 */
/* NOMINAL_SIZE = 12000.0 */
/* NOMINAL_SIZE = 100.0 */
NOMINAL_SIZE = 1500.0
PROCESSING_LEVEL = "L1A"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END
```

A.3.21 MISR ESDT MISBR ODL

```
DATA_TYPE_NAME = "MISBR"
DATA_TYPE_VERSION = "002"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Browse data for use with systematic QA analysis"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 3.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END
```

A.3.22 MISR ESDT MISQA ODL

```
DATA_TYPE_NAME = "MISQA"
DATA_TYPE_VERSION = "002"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Quality Assurance data"
PROVIDER = "Langley Research Center"
/*      Increased to 20.0 from 1.0 by Jim Galasso 10/9/1999      */
NOMINAL_SIZE = 20.0
/*      Changed to Processing Level all 10/9/1999      */
PROCESSING_LEVEL = "ALL"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
PERIOD = "ORBITS=1"
BOUNDARY = "START_OF_ORBIT"
DURATION = "HOURS=2"
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END
```

A.3.23 MISR ESDT MI1B2T ODL

```
DATA_TYPE_NAME = "MI1B2T"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Level 1B2 Terrain Data"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 400.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
DELAY = 1
SPATIAL_FLAG = "Y"
DISTINCT_PARAMETER = "AssociatedSensorShortName"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"
OBJECT = METADATA_DEFINITION
    CLASS = 2
    PARM_NAME = "AssociatedSensorShortName"
    CONTAINER_NAME = "AssociatedPlatformInstrumentSensor"
    TYPE = "STR"
END_OBJECT = METADATA_DEFINITION

END
```

A.3.24 MISR ESDT MI1B2E ODL

```
DATA_TYPE_NAME = "MI1B2E"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Level 1B2 Ellipsoid Data"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 700.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
DELAY = 1
SPATIAL_FLAG = "Y"
DISTINCT_PARAMETER = "AssociatedSensorShortName"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"
OBJECT = METADATA_DEFINITION
    CLASS = 2
```

```

    PARM_NAME = "AssociatedSensorShortName"
    CONTAINER_NAME = "AssociatedPlatformInstrumentSensor"
    TYPE = "STR"
END_OBJECT = METADATA_DEFINITION

END

```

A.3.25 MISR ESDT MIRCCM ODL

```

DATA_TYPE_NAME = "MIRCCM"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR radiometric camera-by-camera Cloud Mask"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 3.0
PROCESSING_LEVEL = "L1B2"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
DELAY = 1
SPATIAL_FLAG = "Y"
DISTINCT_PARAMETER = "AssociatedSensorShortName"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"
OBJECT = METADATA_DEFINITION
    CLASS = 2
    PARM_NAME = "AssociatedSensorShortName"
    CONTAINER_NAME = "AssociatedPlatformInstrumentSensor"
    TYPE = "STR"
END_OBJECT = METADATA_DEFINITION

END

```

A.3.26 MISR ESDT MI1B1 ODL

```

DATA_TYPE_NAME = "MI1B1"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Level 1B2 Ellipsoid Data"
PROVIDER = "Langley Research Center"
/* NOMINAL_SIZE = 574.0 */
/* changed for FILEWATCHER! */
/* NOMINAL_SIZE = 12000.0 */
/* NOMINAL_SIZE = 100.0 */
NOMINAL_SIZE = 1500.0
PROCESSING_LEVEL = "L1B1"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"

```

```

DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END

```

A.3.27 MISR ESDT MIB1LM ODL

```

DATA_TYPE_NAME = "MIB1LM"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MISR"
PLATFORM = "AM1"
DATA_TYPE_DESCRIPTION = "MISR Level 1B1 Local Mode Radiance Data"
PROVIDER = "Langley Research Center"
NOMINAL_SIZE = 100.0
PROCESSING_LEVEL = "L1"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "LARC"
DELAY = 1
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "LARC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "LARC"
PROCESSED_AT = "LARC"

END

```

A.3.28 MISR ORBIT ODL

```

PLATFORM = "AM1"
OBJECT = ORBIT_MODEL
    CLASS = 1

/*  ORBIT_NUMBER = 7472          */
    ORBIT_NUMBER = 7496

/*  Cross correlate using PATHMAP_MISR.odl file */

/*  ORBIT_PATH_NUMBER = 28      */
    ORBIT_PATH_NUMBER = 52

/*  ORBIT_PERIOD = "SECS=5934"  */
    ORBIT_PERIOD = "SECS=5933"

/*  ORBIT_START = "05/14/2001 10:41:32" */

```

```

        ORBIT_START = "05/16/2001 02:14:44"

END_OBJECT = ORBIT_MODEL

END

```

A.3.29 MISR PATHMAP ODL

```

PLATFORM = "AM1"
PATHMAP_NAME = "MISR"
OBJECT = PATHMAP_ENTRY
    CLASS = 1
    ABSOLUTE_PATH = 1
    MAPPED_PATH = 1
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 2
    ABSOLUTE_PATH = 2
    MAPPED_PATH = 17
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 3
    ABSOLUTE_PATH = 3
    MAPPED_PATH = 33
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 4
    ABSOLUTE_PATH = 4
    MAPPED_PATH = 49
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 5
    ABSOLUTE_PATH = 5
    MAPPED_PATH = 65
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 6
    ABSOLUTE_PATH = 6
    MAPPED_PATH = 81
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 7
    ABSOLUTE_PATH = 7
    MAPPED_PATH = 97
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 8
    ABSOLUTE_PATH = 8
    MAPPED_PATH = 113
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 9
    ABSOLUTE_PATH = 9
    MAPPED_PATH = 129
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY

```



```

        CLASS = 10
        ABSOLUTE_PATH = 10
        MAPPED_PATH = 145
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 11
        ABSOLUTE_PATH = 11
        MAPPED_PATH = 161
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 12
        ABSOLUTE_PATH = 12
        MAPPED_PATH = 177
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 13
        ABSOLUTE_PATH = 13
        MAPPED_PATH = 193
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 14
        ABSOLUTE_PATH = 14
        MAPPED_PATH = 209
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 15
        ABSOLUTE_PATH = 15
        MAPPED_PATH = 225
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 16
        ABSOLUTE_PATH = 16
        MAPPED_PATH = 8
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 17
        ABSOLUTE_PATH = 17
        MAPPED_PATH = 24
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 18
        ABSOLUTE_PATH = 18
        MAPPED_PATH = 40
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 19
        ABSOLUTE_PATH = 19
        MAPPED_PATH = 56
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 20
        ABSOLUTE_PATH = 20
        MAPPED_PATH = 72
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 21
        ABSOLUTE_PATH = 21
        MAPPED_PATH = 88

```

```

END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 22
    ABSOLUTE_PATH = 22
    MAPPED_PATH = 104
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 23
    ABSOLUTE_PATH = 23
    MAPPED_PATH = 120
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 24
    ABSOLUTE_PATH = 24
    MAPPED_PATH = 136
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 25
    ABSOLUTE_PATH = 25
    MAPPED_PATH = 152
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 26
    ABSOLUTE_PATH = 26
    MAPPED_PATH = 168
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 27
    ABSOLUTE_PATH = 27
    MAPPED_PATH = 184
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 28
    ABSOLUTE_PATH = 28
    MAPPED_PATH = 200
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 29
    ABSOLUTE_PATH = 29
    MAPPED_PATH = 216
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 30
    ABSOLUTE_PATH = 30
    MAPPED_PATH = 232
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 31
    ABSOLUTE_PATH = 31
    MAPPED_PATH = 15
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 32
    ABSOLUTE_PATH = 32
    MAPPED_PATH = 31
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 33

```

```

        ABSOLUTE_PATH = 33
        MAPPED_PATH = 47
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 34
        ABSOLUTE_PATH = 34
        MAPPED_PATH = 63
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 35
        ABSOLUTE_PATH = 35
        MAPPED_PATH = 79
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 36
        ABSOLUTE_PATH = 36
        MAPPED_PATH = 95
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 37
        ABSOLUTE_PATH = 37
        MAPPED_PATH = 111
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 38
        ABSOLUTE_PATH = 38
        MAPPED_PATH = 127
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 39
        ABSOLUTE_PATH = 39
        MAPPED_PATH = 143
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 40
        ABSOLUTE_PATH = 40
        MAPPED_PATH = 159
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 41
        ABSOLUTE_PATH = 41
        MAPPED_PATH = 175
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 42
        ABSOLUTE_PATH = 42
        MAPPED_PATH = 191
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 43
        ABSOLUTE_PATH = 43
        MAPPED_PATH = 207
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 44
        ABSOLUTE_PATH = 44
        MAPPED_PATH = 223
    END_OBJECT = PATHMAP_ENTRY

```

```

OBJECT = PATHMAP_ENTRY
  CLASS = 45
  ABSOLUTE_PATH = 45
  MAPPED_PATH = 6
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 46
  ABSOLUTE_PATH = 46
  MAPPED_PATH = 22
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 47
  ABSOLUTE_PATH = 47
  MAPPED_PATH = 38
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 48
  ABSOLUTE_PATH = 48
  MAPPED_PATH = 54
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 49
  ABSOLUTE_PATH = 49
  MAPPED_PATH = 70
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 50
  ABSOLUTE_PATH = 50
  MAPPED_PATH = 86
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 51
  ABSOLUTE_PATH = 51
  MAPPED_PATH = 102
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 52
  ABSOLUTE_PATH = 52
  MAPPED_PATH = 118
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 53
  ABSOLUTE_PATH = 53
  MAPPED_PATH = 134
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 54
  ABSOLUTE_PATH = 54
  MAPPED_PATH = 150
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 55
  ABSOLUTE_PATH = 55
  MAPPED_PATH = 166
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 56
  ABSOLUTE_PATH = 56

```

```

        MAPPED_PATH = 182
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 57
        ABSOLUTE_PATH = 57
        MAPPED_PATH = 198
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 58
        ABSOLUTE_PATH = 58
        MAPPED_PATH = 214
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 59
        ABSOLUTE_PATH = 59
        MAPPED_PATH = 230
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 60
        ABSOLUTE_PATH = 60
        MAPPED_PATH = 13
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 61
        ABSOLUTE_PATH = 61
        MAPPED_PATH = 29
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 62
        ABSOLUTE_PATH = 62
        MAPPED_PATH = 45
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 63
        ABSOLUTE_PATH = 63
        MAPPED_PATH = 61
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 64
        ABSOLUTE_PATH = 64
        MAPPED_PATH = 77
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 65
        ABSOLUTE_PATH = 65
        MAPPED_PATH = 93
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 66
        ABSOLUTE_PATH = 66
        MAPPED_PATH = 109
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 67
        ABSOLUTE_PATH = 67
        MAPPED_PATH = 125
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY

```

```

        CLASS = 68
        ABSOLUTE_PATH = 68
        MAPPED_PATH = 141
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 69
        ABSOLUTE_PATH = 69
        MAPPED_PATH = 157
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 70
        ABSOLUTE_PATH = 70
        MAPPED_PATH = 173
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 71
        ABSOLUTE_PATH = 71
        MAPPED_PATH = 189
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 72
        ABSOLUTE_PATH = 72
        MAPPED_PATH = 205
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 73
        ABSOLUTE_PATH = 73
        MAPPED_PATH = 221
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 74
        ABSOLUTE_PATH = 74
        MAPPED_PATH = 4
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 75
        ABSOLUTE_PATH = 75
        MAPPED_PATH = 20
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 76
        ABSOLUTE_PATH = 76
        MAPPED_PATH = 36
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 77
        ABSOLUTE_PATH = 77
        MAPPED_PATH = 52
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 78
        ABSOLUTE_PATH = 78
        MAPPED_PATH = 68
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 79
        ABSOLUTE_PATH = 79
        MAPPED_PATH = 84

```

```

END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 80
    ABSOLUTE_PATH = 80
    MAPPED_PATH = 100
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 81
    ABSOLUTE_PATH = 81
    MAPPED_PATH = 116
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 82
    ABSOLUTE_PATH = 82
    MAPPED_PATH = 132
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 83
    ABSOLUTE_PATH = 83
    MAPPED_PATH = 148
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 84
    ABSOLUTE_PATH = 84
    MAPPED_PATH = 164
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 85
    ABSOLUTE_PATH = 85
    MAPPED_PATH = 180
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 86
    ABSOLUTE_PATH = 86
    MAPPED_PATH = 196
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 87
    ABSOLUTE_PATH = 87
    MAPPED_PATH = 212
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 88
    ABSOLUTE_PATH = 88
    MAPPED_PATH = 228
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 89
    ABSOLUTE_PATH = 89
    MAPPED_PATH = 11
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 90
    ABSOLUTE_PATH = 90
    MAPPED_PATH = 27
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 91

```

```

        ABSOLUTE_PATH = 91
        MAPPED_PATH = 43
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 92
        ABSOLUTE_PATH = 92
        MAPPED_PATH = 59
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 93
        ABSOLUTE_PATH = 93
        MAPPED_PATH = 75
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 94
        ABSOLUTE_PATH = 94
        MAPPED_PATH = 91
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 95
        ABSOLUTE_PATH = 95
        MAPPED_PATH = 107
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 96
        ABSOLUTE_PATH = 96
        MAPPED_PATH = 123
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 97
        ABSOLUTE_PATH = 97
        MAPPED_PATH = 139
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 98
        ABSOLUTE_PATH = 98
        MAPPED_PATH = 155
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 99
        ABSOLUTE_PATH = 99
        MAPPED_PATH = 171
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 100
        ABSOLUTE_PATH = 100
        MAPPED_PATH = 187
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 101
        ABSOLUTE_PATH = 101
        MAPPED_PATH = 203
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 102
        ABSOLUTE_PATH = 102
        MAPPED_PATH = 219
    END_OBJECT = PATHMAP_ENTRY

```



```

OBJECT = PATHMAP_ENTRY
  CLASS = 103
  ABSOLUTE_PATH = 103
  MAPPED_PATH = 2
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 104
  ABSOLUTE_PATH = 104
  MAPPED_PATH = 18
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 105
  ABSOLUTE_PATH = 105
  MAPPED_PATH = 34
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 106
  ABSOLUTE_PATH = 106
  MAPPED_PATH = 50
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 107
  ABSOLUTE_PATH = 107
  MAPPED_PATH = 66
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 108
  ABSOLUTE_PATH = 108
  MAPPED_PATH = 82
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 109
  ABSOLUTE_PATH = 109
  MAPPED_PATH = 98
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 110
  ABSOLUTE_PATH = 110
  MAPPED_PATH = 114
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 111
  ABSOLUTE_PATH = 111
  MAPPED_PATH = 130
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 112
  ABSOLUTE_PATH = 112
  MAPPED_PATH = 146
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 113
  ABSOLUTE_PATH = 113
  MAPPED_PATH = 162
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 114
  ABSOLUTE_PATH = 114

```

```

        MAPPED_PATH = 178
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 115
        ABSOLUTE_PATH = 115
        MAPPED_PATH = 194
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 116
        ABSOLUTE_PATH = 116
        MAPPED_PATH = 210
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 117
        ABSOLUTE_PATH = 117
        MAPPED_PATH = 226
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 118
        ABSOLUTE_PATH = 118
        MAPPED_PATH = 9
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 119
        ABSOLUTE_PATH = 119
        MAPPED_PATH = 25
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 120
        ABSOLUTE_PATH = 120
        MAPPED_PATH = 41
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 121
        ABSOLUTE_PATH = 121
        MAPPED_PATH = 57
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 122
        ABSOLUTE_PATH = 122
        MAPPED_PATH = 73
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 123
        ABSOLUTE_PATH = 123
        MAPPED_PATH = 89
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 124
        ABSOLUTE_PATH = 124
        MAPPED_PATH = 105
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 125
        ABSOLUTE_PATH = 125
        MAPPED_PATH = 121
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY

```

```

        CLASS = 126
        ABSOLUTE_PATH = 126
        MAPPED_PATH = 137
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 127
        ABSOLUTE_PATH = 127
        MAPPED_PATH = 153
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 128
        ABSOLUTE_PATH = 128
        MAPPED_PATH = 169
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 129
        ABSOLUTE_PATH = 129
        MAPPED_PATH = 185
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 130
        ABSOLUTE_PATH = 130
        MAPPED_PATH = 201
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 131
        ABSOLUTE_PATH = 131
        MAPPED_PATH = 217
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 132
        ABSOLUTE_PATH = 132
        MAPPED_PATH = 233
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 133
        ABSOLUTE_PATH = 133
        MAPPED_PATH = 16
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 134
        ABSOLUTE_PATH = 134
        MAPPED_PATH = 32
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 135
        ABSOLUTE_PATH = 135
        MAPPED_PATH = 48
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 136
        ABSOLUTE_PATH = 136
        MAPPED_PATH = 64
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 137
        ABSOLUTE_PATH = 137
        MAPPED_PATH = 80

```

```

END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 138
    ABSOLUTE_PATH = 138
    MAPPED_PATH = 96
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 139
    ABSOLUTE_PATH = 139
    MAPPED_PATH = 112
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 140
    ABSOLUTE_PATH = 140
    MAPPED_PATH = 128
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 141
    ABSOLUTE_PATH = 141
    MAPPED_PATH = 144
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 142
    ABSOLUTE_PATH = 142
    MAPPED_PATH = 160
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 143
    ABSOLUTE_PATH = 143
    MAPPED_PATH = 176
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 144
    ABSOLUTE_PATH = 144
    MAPPED_PATH = 192
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 145
    ABSOLUTE_PATH = 145
    MAPPED_PATH = 208
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 146
    ABSOLUTE_PATH = 146
    MAPPED_PATH = 224
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 147
    ABSOLUTE_PATH = 147
    MAPPED_PATH = 7
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 148
    ABSOLUTE_PATH = 148
    MAPPED_PATH = 23
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 149

```

```

        ABSOLUTE_PATH = 149
        MAPPED_PATH = 39
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 150
        ABSOLUTE_PATH = 150
        MAPPED_PATH = 55
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 151
        ABSOLUTE_PATH = 151
        MAPPED_PATH = 71
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 152
        ABSOLUTE_PATH = 152
        MAPPED_PATH = 87
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 153
        ABSOLUTE_PATH = 153
        MAPPED_PATH = 103
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 154
        ABSOLUTE_PATH = 154
        MAPPED_PATH = 119
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 155
        ABSOLUTE_PATH = 155
        MAPPED_PATH = 135
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 156
        ABSOLUTE_PATH = 156
        MAPPED_PATH = 151
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 157
        ABSOLUTE_PATH = 157
        MAPPED_PATH = 167
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 158
        ABSOLUTE_PATH = 158
        MAPPED_PATH = 183
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 159
        ABSOLUTE_PATH = 159
        MAPPED_PATH = 199
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 160
        ABSOLUTE_PATH = 160
        MAPPED_PATH = 215
    END_OBJECT = PATHMAP_ENTRY

```

```

OBJECT = PATHMAP_ENTRY
  CLASS = 161
  ABSOLUTE_PATH = 161
  MAPPED_PATH = 231
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 162
  ABSOLUTE_PATH = 162
  MAPPED_PATH = 14
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 163
  ABSOLUTE_PATH = 163
  MAPPED_PATH = 30
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 164
  ABSOLUTE_PATH = 164
  MAPPED_PATH = 46
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 165
  ABSOLUTE_PATH = 165
  MAPPED_PATH = 62
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 166
  ABSOLUTE_PATH = 166
  MAPPED_PATH = 78
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 167
  ABSOLUTE_PATH = 167
  MAPPED_PATH = 94
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 168
  ABSOLUTE_PATH = 168
  MAPPED_PATH = 110
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 169
  ABSOLUTE_PATH = 169
  MAPPED_PATH = 126
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 170
  ABSOLUTE_PATH = 170
  MAPPED_PATH = 142
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 171
  ABSOLUTE_PATH = 171
  MAPPED_PATH = 158
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 172
  ABSOLUTE_PATH = 172

```

```

        MAPPED_PATH = 174
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 173
        ABSOLUTE_PATH = 173
        MAPPED_PATH = 190
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 174
        ABSOLUTE_PATH = 174
        MAPPED_PATH = 206
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 175
        ABSOLUTE_PATH = 175
        MAPPED_PATH = 222
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 176
        ABSOLUTE_PATH = 176
        MAPPED_PATH = 5
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 177
        ABSOLUTE_PATH = 177
        MAPPED_PATH = 21
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 178
        ABSOLUTE_PATH = 178
        MAPPED_PATH = 37
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 179
        ABSOLUTE_PATH = 179
        MAPPED_PATH = 53
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 180
        ABSOLUTE_PATH = 180
        MAPPED_PATH = 69
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 181
        ABSOLUTE_PATH = 181
        MAPPED_PATH = 85
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 182
        ABSOLUTE_PATH = 182
        MAPPED_PATH = 101
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 183
        ABSOLUTE_PATH = 183
        MAPPED_PATH = 117
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY

```

```

        CLASS = 184
        ABSOLUTE_PATH = 184
        MAPPED_PATH = 133
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 185
        ABSOLUTE_PATH = 185
        MAPPED_PATH = 149
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 186
        ABSOLUTE_PATH = 186
        MAPPED_PATH = 165
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 187
        ABSOLUTE_PATH = 187
        MAPPED_PATH = 181
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 188
        ABSOLUTE_PATH = 188
        MAPPED_PATH = 197
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 189
        ABSOLUTE_PATH = 189
        MAPPED_PATH = 213
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 190
        ABSOLUTE_PATH = 190
        MAPPED_PATH = 229
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 191
        ABSOLUTE_PATH = 191
        MAPPED_PATH = 12
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 192
        ABSOLUTE_PATH = 192
        MAPPED_PATH = 28
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 193
        ABSOLUTE_PATH = 193
        MAPPED_PATH = 44
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 194
        ABSOLUTE_PATH = 194
        MAPPED_PATH = 60
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 195
        ABSOLUTE_PATH = 195
        MAPPED_PATH = 76

```



```

END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 196
    ABSOLUTE_PATH = 196
    MAPPED_PATH = 92
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 197
    ABSOLUTE_PATH = 197
    MAPPED_PATH = 108
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 198
    ABSOLUTE_PATH = 198
    MAPPED_PATH = 124
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 199
    ABSOLUTE_PATH = 199
    MAPPED_PATH = 140
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 200
    ABSOLUTE_PATH = 200
    MAPPED_PATH = 156
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 201
    ABSOLUTE_PATH = 201
    MAPPED_PATH = 172
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 202
    ABSOLUTE_PATH = 202
    MAPPED_PATH = 188
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 203
    ABSOLUTE_PATH = 203
    MAPPED_PATH = 204
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 204
    ABSOLUTE_PATH = 204
    MAPPED_PATH = 220
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 205
    ABSOLUTE_PATH = 205
    MAPPED_PATH = 3
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 206
    ABSOLUTE_PATH = 206
    MAPPED_PATH = 19
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
    CLASS = 207

```

```

        ABSOLUTE_PATH = 207
        MAPPED_PATH = 35
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 208
        ABSOLUTE_PATH = 208
        MAPPED_PATH = 51
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 209
        ABSOLUTE_PATH = 209
        MAPPED_PATH = 67
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 210
        ABSOLUTE_PATH = 210
        MAPPED_PATH = 83
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 211
        ABSOLUTE_PATH = 211
        MAPPED_PATH = 99
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 212
        ABSOLUTE_PATH = 212
        MAPPED_PATH = 115
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 213
        ABSOLUTE_PATH = 213
        MAPPED_PATH = 131
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 214
        ABSOLUTE_PATH = 214
        MAPPED_PATH = 147
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 215
        ABSOLUTE_PATH = 215
        MAPPED_PATH = 163
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 216
        ABSOLUTE_PATH = 216
        MAPPED_PATH = 179
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 217
        ABSOLUTE_PATH = 217
        MAPPED_PATH = 195
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 218
        ABSOLUTE_PATH = 218
        MAPPED_PATH = 211
    END_OBJECT = PATHMAP_ENTRY

```

```

OBJECT = PATHMAP_ENTRY
  CLASS = 219
  ABSOLUTE_PATH = 219
  MAPPED_PATH = 227
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 220
  ABSOLUTE_PATH = 220
  MAPPED_PATH = 10
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 221
  ABSOLUTE_PATH = 221
  MAPPED_PATH = 26
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 222
  ABSOLUTE_PATH = 222
  MAPPED_PATH = 42
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 223
  ABSOLUTE_PATH = 223
  MAPPED_PATH = 58
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 224
  ABSOLUTE_PATH = 224
  MAPPED_PATH = 74
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 225
  ABSOLUTE_PATH = 225
  MAPPED_PATH = 90
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 226
  ABSOLUTE_PATH = 226
  MAPPED_PATH = 106
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 227
  ABSOLUTE_PATH = 227
  MAPPED_PATH = 122
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 228
  ABSOLUTE_PATH = 228
  MAPPED_PATH = 138
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 229
  ABSOLUTE_PATH = 229
  MAPPED_PATH = 154
END_OBJECT = PATHMAP_ENTRY
OBJECT = PATHMAP_ENTRY
  CLASS = 230
  ABSOLUTE_PATH = 230

```

```

        MAPPED_PATH = 170
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 231
        ABSOLUTE_PATH = 231
        MAPPED_PATH = 186
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 232
        ABSOLUTE_PATH = 232
        MAPPED_PATH = 202
    END_OBJECT = PATHMAP_ENTRY
    OBJECT = PATHMAP_ENTRY
        CLASS = 233
        ABSOLUTE_PATH = 233
        MAPPED_PATH = 218
    END_OBJECT = PATHMAP_ENTRY
END

```

A.4 Typical Terra MODIS PGE & ESDT ODL Files

Listings are provided for the following MODIS ODL files:

A.4.1 MODIS PGE ODL for PGE_NAME PGE01

A.4.2 MODIS ESDT MOD000 ODL

A.4.3 MODIS ESDT MOD01 ODL

A.4.4 MODIS ESDT MOD01LUT ODL

A.4.5 MODIS ESDT MOD03 ODL

A.4.6 MODIS ESDT MOD03LUT ODL

A.4.7 MODIS PGE ODL for PGE_NAME PGE03

A.4.8 GDAS_0ZF ODL

A.4.9 OZ_DAILY ODL

A.4.10 REYNSST ODL

A.4.11 SEA_ICE ODL

A.4.12 NISE ODL

A typical MODIS PGE will differ from the examples here by the PGE_NAME, the specific input/output files referenced, and runtime parameters. However, the overall structure of a given MODIS PGE ODL file would be similar to the ones used here. (N.B. The ODL files shown here are associated with the MODIS version 2.1 software)

A.4.1 MODIS PGE PGE01 ODL

```
PGE_NAME = "PGE01"
PGE_VERSION = "2.1"
PROFILE_ID = 1
PROFILE_DESCRIPTION = "The profile for MOD_PR01 and MOD_PR03 "
PLATFORM = "AM1"
INSTRUMENT = "MODIS"
MINIMUM_OUTPUTS = 0
SCHEDULE_TYPE = "Time"
PROCESSING_PERIOD = "MINS=15"
PROCESSING_BOUNDARY = "START_OF_MIN"
PGE_SSW_VERSION = "2.1"
QUERY_DELAY = 0

OBJECT = EXIT_MESSAGE
  CLASS= 1
  EXIT_CODE = 0
  EXIT_MESSAGE = "PGE01 Exit"
END_OBJECT = EXIT_MESSAGE
OBJECT = EXIT_DEPENDENCY
  CLASS= 1
  DEPENDENCY_PGE_NAME = "none"
  DEPENDENCY_SSW_VERSION = "none"
  EXIT_OPERATION = "="
  EXIT_CODE = 0
END_OBJECT = EXIT_DEPENDENCY

OBJECT = PCF_ENTRY
  CLASS = 10
  LOGICAL_ID = 599001
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "MOD000"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 1
  BEGIN_PERIOD_OFFSET = -7200
  END_PERIOD_OFFSET = -7200
  INPUT_TYPE = "Optional"
  ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"
  OBJECT = FILETYPE
    FILETYPE_NAME = "L0 Data Files"
    CLASS = 1
  END_OBJECT = FILETYPE
  OBJECT = OPTIONAL_INPUT
    CLASS = 1
    ORDER = 1
    RUNTIME_PARM_ID = 51
    TIMER = "HOURS=4"
    TEMPORAL = "N"
  END_OBJECT = OPTIONAL_INPUT
END_OBJECT = PCF_ENTRY
```

```

OBJECT = PCF_ENTRY
  CLASS = 11
  LOGICAL_ID = 599002
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "MOD000"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 1
  BEGIN_PERIOD_OFFSET = 0
  END_PERIOD_OFFSET = 0
  INPUT_TYPE = "Required"
  ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"
  OBJECT = FILETYPE
    FILETYPE_NAME = "L0 Data Files"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
  CLASS = 12
  LOGICAL_ID = 599003
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "MOD01LUT"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 1
  BEGIN_PERIOD_OFFSET = 0
  END_PERIOD_OFFSET = 0
  SCIENCE_GROUP = "L1"
  INPUT_TYPE = "Required"
  ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
  CLASS = 15
  LOGICAL_ID = 600020
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "MOD01LUT"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 1
  BEGIN_PERIOD_OFFSET = 0
  END_PERIOD_OFFSET = 0
  SCIENCE_GROUP = "L2"
  INPUT_TYPE = "Required"
  ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"

```

```

NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
OBJECT = FILETYPE
    FILETYPE_NAME = "Two GEO_parameter data files"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 17
    LOGICAL_ID = 10501
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "AM1EPHN0"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = 0
    END_PERIOD_OFFSET = 0
    INPUT_TYPE = "Required"
    ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 18
    LOGICAL_ID = 10502
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "AM1ATTN0"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = 0
    END_PERIOD_OFFSET = 0
    INPUT_TYPE = "Required"
    ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 110
    LOGICAL_ID = 500100
    PCF_FILE_TYPE = 2

```

```

DATA_TYPE_NAME = "MOD01"
DATA_TYPE_VERSION = "001"
MIN_GRANULE_YIELD = 3
MAX_GRANULE_YIELD = 3
ASSOCIATED_MCF_ID = 500500
SCIENCE_GROUP = "S1"
INSTANCE = 0
MINIMUM_SIZE = 0
MAXIMUM_SIZE = 0
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
CLASS = 111
LOGICAL_ID = 600000
PCF_FILE_TYPE = 2
DATA_TYPE_NAME = "MOD03"
DATA_TYPE_VERSION = "001"
MIN_GRANULE_YIELD = 3
MAX_GRANULE_YIELD = 3
ASSOCIATED_MCF_ID = 600111
SCIENCE_GROUP = "S2"
INSTANCE = 0
MINIMUM_SIZE = 0
MAXIMUM_SIZE = 0
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
CLASS = 113
LOGICAL_ID = 503000
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Length of L1A granules in seconds"
PGE_PARAMETER_DEFAULT = "300.000000"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
CLASS = 114
LOGICAL_ID = 504000
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Scan rate for L1A granule"
PGE_PARAMETER_DEFAULT = "1.477"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
CLASS = 115
LOGICAL_ID = 505000
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "PGE version for L1A granule"
PGE_PARAMETER_DEFAULT = "2.1.1"

```



```

    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 116
    LOGICAL_ID = 800510
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "SatelliteInstrument; AM1M-Terra, PM1M-Aqua"
    PGE_PARAMETER_DEFAULT = "AM1M"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 117
    LOGICAL_ID = 800500
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "PGE01 Version"
    PGE_PARAMETER_DEFAULT = "2.1.1"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 118
    LOGICAL_ID = 600280
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Source for spacecraft kinematic state"
    PGE_PARAMETER_DEFAULT = "SDP Toolkit"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 119
    LOGICAL_ID = 600310
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Terrain Correction Flag"
    PGE_PARAMETER_DEFAULT = "TRUE"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 120
    LOGICAL_ID = 600001
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "LOCALVERSIONID"
    PGE_PARAMETER_DEFAULT = "2.1.2"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

END

```

A.4.2 MODIS ESDT MOD000 ODL

```

DATA_TYPE_NAME = "MOD000"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MODIS"
PLATFORM = "EOSAM1"

```

```

DATA_TYPE_DESCRIPTION = "L0 Input of PGE MOD_PR01"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 569.0
PROCESSING_LEVEL = "L0"
HDF_DATA = "N"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "GSFC"
PERIOD = "HOURS=2"
BOUNDARY = "START_OF_DAY"
DELAY = 43200
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
OBJECT = FILETYPE
    CLASS = 1
    FILETYPE_NAME = "L0 Data Files"
    MAXIMUM_NUM_FILES = 6
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
END

```

A.4.3 MODIS ESDT MOD01 ODL

```

DATA_TYPE_NAME = "MOD01"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MODIS"
PLATFORM = "EOSAM1"
DATA_TYPE_DESCRIPTION = "An Input of PGE MOD_PR02"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 569.0
PROCESSING_LEVEL = "L1A"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "GSFC"
PERIOD = "MINS=5"
BOUNDARY = "START_OF_MIN"
DELAY = 43200
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
END

```

A.4.4 MODIS ESDT MOD01LUT ODL

```

DATA_TYPE_NAME = "MOD01LUT"

```

```

DATA_TYPE_VERSION = "001"
INSTRUMENT = "MODIS"
PLATFORM = "EOSAM1"
DATA_TYPE_DESCRIPTION = "An Input (static) of PGE MOD01"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 0.357
PROCESSING_LEVEL = "L1"
HDF_DATA = "N"
DYNAMIC_FLAG = "S"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
END

```

A.4.5 MODIS ESDT MOD03 ODL

```

DATA_TYPE_NAME = "MOD03"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MODIS"
PLATFORM = "EOSAM1"
DATA_TYPE_DESCRIPTION = "Input/Output of PGE MOD_PR29/MOD_PR03"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 58.0
PROCESSING_LEVEL = "Geo"
HDF_DATA = "Y"
DYNAMIC_FLAG = "I"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "GSFC"
PERIOD = "MINS=5"
BOUNDARY = "START_OF_MIN"
DELAY = 43200
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
END

```

A.4.6 MODIS ESDT MOD03LUT ODL

```

DATA_TYPE_NAME = "MOD03LUT"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MODIS"
PLATFORM = "EOSAM1"
DATA_TYPE_DESCRIPTION = "An Input (static) of PGE MOD_PR03"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 0.357
PROCESSING_LEVEL = "L1"
HDF_DATA = "N"
DYNAMIC_FLAG = "S"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT

```

```

        CLASS = 1
        USED_BY = "GSFC"
    END_OBJECT = USE_OBJECT
    OBJECT = FILETYPE
        CLASS = 1
        FILETYPE_NAME = "Two GEO_parameter data files"
        MAXIMUM_NUM_FILES = 2
    END_OBJECT = FILETYPE
END

```

A.4.7 MODIS PGE PGE03 ODL

```

PGE_NAME = "TerraPGE03"
PGE_VERSION = "3.0.0"
PROFILE_ID = 1
PROFILE_DESCRIPTION = "First Step in Level 2 Processing"
PGE_DEFAULT_PROFILE = "N"
PLATFORM = "AM1"
INSTRUMENT = "MODIS"
MINIMUM_OUTPUTS = 0
SCHEDULE_TYPE = "Time"
PROCESSING_PERIOD = "MINS=5"
PROCESSING_BOUNDARY = "START_OF_MIN"
PGE_SSW_VERSION = "3.0.0"
QUERY_DELAY = 0
OBJECT = EXIT_MESSAGE
    CLASS= 1
    EXIT_CODE = 0
    EXIT_MESSAGE = "none"
END_OBJECT = EXIT_MESSAGE
OBJECT = EXIT_DEPENDENCY
    CLASS= 1
    DEPENDENCY_PGE_NAME = "none"
    DEPENDENCY_SSW_VERSION = "none"
    EXIT_OPERATION = "="
    EXIT_CODE = 0
END_OBJECT = EXIT_DEPENDENCY
OBJECT = PCF_ENTRY
    CLASS = 11
    LOGICAL_ID = 600000
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MOD03"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = 0
    END_PERIOD_OFFSET = 0
    INPUT_TYPE = "Required"
    ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

```

```

OBJECT = PCF_ENTRY
  CLASS = 12
  LOGICAL_ID = 700000
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "MOD02QKM"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 1
  BEGIN_PERIOD_OFFSET = 0
  END_PERIOD_OFFSET = 0
  INPUT_TYPE = "Required"
  ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 13
  LOGICAL_ID = 700002
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "MOD021KM"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 1
  BEGIN_PERIOD_OFFSET = 0
  END_PERIOD_OFFSET = 0
  INPUT_TYPE = "Required"
  ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 21
  LOGICAL_ID = 900000
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "GDAS_0ZF"
  DATA_TYPE_VERSION = "001"
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 1
  BEGIN_PERIOD_OFFSET = -10650
  END_PERIOD_OFFSET = 10650
  INPUT_TYPE = "Required"
  ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  SPATIAL_TIME_DELTA = 0
  KEY_INPUT = "N"

```

```

OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 22
    LOGICAL_ID = 900020
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "OZ_DAILY"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = -43200
    END_PERIOD_OFFSET = 43200
    INPUT_TYPE = "Required"
    ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 23
    LOGICAL_ID = 900030
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "REYNSST"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = 0
    END_PERIOD_OFFSET = 0
    INPUT_TYPE = "Required"
    ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 24
    LOGICAL_ID = 900040
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "SEA_ICE"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = -43200
    END_PERIOD_OFFSET = 43200
    INPUT_TYPE = "Required"

```

```

ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 25
    LOGICAL_ID = 900100
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "NISE"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = 0
    END_PERIOD_OFFSET = 0
    INPUT_TYPE = "Required"
    ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 31
    LOGICAL_ID = 420011
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MOD07LUT"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = 0
    END_PERIOD_OFFSET = 0
    SCIENCE_GROUP = "L1"
    INPUT_TYPE = "Required"
    ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 32
    LOGICAL_ID = 420012
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MOD07LUT"

```

```

DATA_TYPE_VERSION = "001"
MIN_GRANULES_REQUIRED = 1
MAX_GRANULES_REQUIRED = 1
BEGIN_PERIOD_OFFSET = 0
END_PERIOD_OFFSET = 0
SCIENCE_GROUP = "L2"
INPUT_TYPE = "Required"
ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
SPATIAL_TIME_DELTA = 0
KEY_INPUT = "N"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 33
    LOGICAL_ID = 422501
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MOD35ANC"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = 0
    END_PERIOD_OFFSET = 0
    SCIENCE_GROUP = "L1"
    INPUT_TYPE = "Required"
    ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 34
    LOGICAL_ID = 900600
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MOD35ANC"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = 0
    END_PERIOD_OFFSET = 0
    SCIENCE_GROUP = "L2"
    INPUT_TYPE = "Required"
    ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"

```



```

        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 35
    LOGICAL_ID = 900601
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "MOD35ANC"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    BEGIN_PERIOD_OFFSET = 0
    END_PERIOD_OFFSET = 0
    SCIENCE_GROUP = "L3"
    INPUT_TYPE = "Required"
    ALIGN_DPR_TIME_WITH_INPUT_TIME = "N"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    SPATIAL_TIME_DELTA = 0
    KEY_INPUT = "N"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 122
    LOGICAL_ID = 402500
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MODVOLC"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULE_YIELD = 1
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 402503
    SCIENCE_GROUP = "S1"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 123
    LOGICAL_ID = 420000
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MOD07_L2"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULE_YIELD = 1
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 420001
    SCIENCE_GROUP = "S2"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"

```

```

        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 124
    LOGICAL_ID = 420002
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MOD07_QC"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULE_YIELD = 0
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 420003
    SCIENCE_GROUP = "S3"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 125
    LOGICAL_ID = 422500
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MOD35_L2"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULE_YIELD = 1
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 422506
    SCIENCE_GROUP = "S4"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 126
    LOGICAL_ID = 422551
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "MOD35_QC"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULE_YIELD = 1
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 422507
    SCIENCE_GROUP = "S5"
    INSTANCE = 0
    MINIMUM_SIZE = 0
    MAXIMUM_SIZE = 0
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY

```

```

CLASS = 127
LOGICAL_ID = 422552
PCF_FILE_TYPE = 2
DATA_TYPE_NAME = "MODCSR_G"
DATA_TYPE_VERSION = "001"
MIN_GRANULE_YIELD = 1
MAX_GRANULE_YIELD = 1
ASSOCIATED_MCF_ID = 422510
SCIENCE_GROUP = "S6"
INSTANCE = 0
MINIMUM_SIZE = 0
MAXIMUM_SIZE = 0
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 129
    LOGICAL_ID = 800510
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "SatelliteInstrument"
    PGE_PARAMETER_DEFAULT = "AM1M"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 130
    LOGICAL_ID = 402502
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "RP Reference to VOLCALERT"
    PGE_PARAMETER_DEFAULT = "402500:1"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 131
    LOGICAL_ID = 420004
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "MOD_PR07.qc"
    PGE_PARAMETER_DEFAULT = "420002:1"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 132
    LOGICAL_ID = 421000
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Number_Of_Invent_RP"
    PGE_PARAMETER_DEFAULT = "4"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 133
    LOGICAL_ID = 421001
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Inventory_RP_Name_1 "

```

```

    PGE_PARAMETER_DEFAULT = "REPROCESSINGACTUAL"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 134
    LOGICAL_ID = 421002
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Inventory_RP_Value_1"
    PGE_PARAMETER_DEFAULT = "processed once"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 135
    LOGICAL_ID = 421003
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Inventory_RP_Name_2 "
    PGE_PARAMETER_DEFAULT = "REPROCESSINGPLANNED"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 136
    LOGICAL_ID = 421004
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Inventory_RP_Value_2"
    PGE_PARAMETER_DEFAULT = "further update is anticipated"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 137
    LOGICAL_ID = 421005
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Inventory_RP_Name_3 "
    PGE_PARAMETER_DEFAULT = "LOCALVERSIONID"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 138
    LOGICAL_ID = 421006
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Inventory_RP_Value_3"
    PGE_PARAMETER_DEFAULT = "002"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 139
    LOGICAL_ID = 421007
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Inventory_RP_Name_4 "
    PGE_PARAMETER_DEFAULT = "PGEVERSION"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY

```

```

OBJECT = PCF_ENTRY
  CLASS = 140
  LOGICAL_ID = 421008
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Inventory_RP_Value_4"
  PGE_PARAMETER_DEFAULT = "3.0.0"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 141
  LOGICAL_ID = 421100
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Number_Of_Archive_RP"
  PGE_PARAMETER_DEFAULT = "8"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 142
  LOGICAL_ID = 421101
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Archive_RP_Name_1 "
  PGE_PARAMETER_DEFAULT = "ALGORITHMPACKAGEACCEPTANCEDATE"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 143
  LOGICAL_ID = 421102
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Archive_RP_Value_1"
  PGE_PARAMETER_DEFAULT = "June 1997"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 144
  LOGICAL_ID = 421103
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Archive_RP_Name_2 "
  PGE_PARAMETER_DEFAULT = "ALGORITHMPACKAGEMATURITCODE"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 145
  LOGICAL_ID = 421104
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Archive_RP_Value_2"
  PGE_PARAMETER_DEFAULT = "at-launch"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 146
  LOGICAL_ID = 421105
  PCF_FILE_TYPE = 5

```

```

    PGE_PARAMETER_NAME = "Archive RP_Name_3 "
    PGE_PARAMETER_DEFAULT = "ALGORITHMPACKAGENAME"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 147
    LOGICAL_ID = 421106
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Archive RP_Value_3"
    PGE_PARAMETER_DEFAULT = "ATBD-MOD-07"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 148
    LOGICAL_ID = 421107
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Archive RP_Name_4 "
    PGE_PARAMETER_DEFAULT = "ALGORITHMPACKAGEVERSION"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 149
    LOGICAL_ID = 421108
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Archive RP_Value_4"
    PGE_PARAMETER_DEFAULT = "2"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 150
    LOGICAL_ID = 421109
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Archive RP_Name_5 "
    PGE_PARAMETER_DEFAULT = "INSTRUMENTNAME"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 151
    LOGICAL_ID = 421110
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Archive RP_Value_5"
    PGE_PARAMETER_DEFAULT = "Moderate Resolution Imaging Spectroradiometer"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 152
    LOGICAL_ID = 421111
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Archive RP_Name_6 "
    PGE_PARAMETER_DEFAULT = "Profiles_Algorithm_Version_Number"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"

```

```

END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 153
  LOGICAL_ID = 421112
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Archive_RP_Value_6"
  PGE_PARAMETER_DEFAULT = "1"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 154
  LOGICAL_ID = 421113
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Archive_RP_Name_7 "
  PGE_PARAMETER_DEFAULT = "Total_Ozone_Algorithm_Version_Number"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 155
  LOGICAL_ID = 421114
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Archive_RP_Value_7"
  PGE_PARAMETER_DEFAULT = "1"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 156
  LOGICAL_ID = 421115
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Archive_RP_Name_8 "
  PGE_PARAMETER_DEFAULT = "Stability_Indices_Algorithm_Version_Number"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 157
  LOGICAL_ID = 421116
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Archive_RP_Value_8"
  PGE_PARAMETER_DEFAULT = "1"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 158
  LOGICAL_ID = 422508
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "MOD35_QC.qc"
  PGE_PARAMETER_DEFAULT = "422551:1"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
  PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 159
  LOGICAL_ID = 424000

```

```

PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "MOD35_Num_InvMet_RP_Pairs"
PGE_PARAMETER_DEFAULT = "4"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 160
LOGICAL_ID = 424001
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Inventory_RP_Name_1 "
PGE_PARAMETER_DEFAULT = "REPROCESSINGACTUAL"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 161
LOGICAL_ID = 424002
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Inventory_RP_Value_1"
PGE_PARAMETER_DEFAULT = "processed once"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 162
LOGICAL_ID = 424003
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Inventory_RP_Name_2 "
PGE_PARAMETER_DEFAULT = "REPROCESSINGPLANNED"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 163
LOGICAL_ID = 424004
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Inventory_RP_Value_2"
PGE_PARAMETER_DEFAULT = "further update is anticipated"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 164
LOGICAL_ID = 424005
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Inventory_RP_Name_3 "
PGE_PARAMETER_DEFAULT = "LOCALVERSIONID"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 165
LOGICAL_ID = 424006
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Inventory_RP_Value_3"
PGE_PARAMETER_DEFAULT = "002"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"

```



```

    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 166
    LOGICAL_ID = 424007
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Inventory_RP_Name_4 "
    PGE_PARAMETER_DEFAULT = "PGEVERSION"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 167
    LOGICAL_ID = 424008
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Inventory_RP_Value_4"
    PGE_PARAMETER_DEFAULT = "2.6.1"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 168
    LOGICAL_ID = 424100
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "MOD35_Num_ArchMet_RP_Pairs"
    PGE_PARAMETER_DEFAULT = "5"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 169
    LOGICAL_ID = 424101
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Archive_RP_Name_1 "
    PGE_PARAMETER_DEFAULT = "ALGORITHMPACKAGEACCEPTANCEDATE"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 170
    LOGICAL_ID = 424102
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Archive_RP_Value_1"
    PGE_PARAMETER_DEFAULT = "June 1997"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 171
    LOGICAL_ID = 424103
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Archive_RP_Name_2 "
    PGE_PARAMETER_DEFAULT = "ALGORITHMPACKAGEMATURITYCODE"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
    PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 172

```

```

LOGICAL_ID = 424104
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Archive_RP_Value_2"
PGE_PARAMETER_DEFAULT = "at-launch"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 173
LOGICAL_ID = 424105
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Archive_RP_Name_3 "
PGE_PARAMETER_DEFAULT = "ALGORITHMPACKAGENAME"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 174
LOGICAL_ID = 424106
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Archive_RP_Value_3"
PGE_PARAMETER_DEFAULT = "ATBD-MOD-06"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 175
LOGICAL_ID = 424107
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Archive_RP_Name_4 "
PGE_PARAMETER_DEFAULT = "ALGORITHMPACKAGEVERSION"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 176
LOGICAL_ID = 424108
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Archive_RP_Value_4"
PGE_PARAMETER_DEFAULT = "2"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 177
LOGICAL_ID = 424109
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Archive_RP_Name_5 "
PGE_PARAMETER_DEFAULT = "INSTRUMENTNAME"
PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
PROFILE_SELECTOR_PGE_PARAMETER = "N"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
CLASS = 178
LOGICAL_ID = 424110
PCF_FILE_TYPE = 5
PGE_PARAMETER_NAME = "Archive_RP_Value_5"
PGE_PARAMETER_DEFAULT = "Moderate Resolution Imaging Spectroradiometer"

```

```

        PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
        PROFILE_SELECTOR_PGE_PARAMETER = "N"
    END_OBJECT = PCF_ENTRY
    OBJECT = PCF_ENTRY
        CLASS = 179
        LOGICAL_ID = 424300
        PCF_FILE_TYPE = 5
        PGE_PARAMETER_NAME = "UW DEBUG; 0 to 4, no output to reams"
        PGE_PARAMETER_DEFAULT = "0"
        PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
        PROFILE_SELECTOR_PGE_PARAMETER = "N"
    END_OBJECT = PCF_ENTRY
    OBJECT = PCF_ENTRY
        CLASS = 180
        LOGICAL_ID = 424301
        PCF_FILE_TYPE = 5
        PGE_PARAMETER_NAME = "Processing Range Begin Line"
        PGE_PARAMETER_DEFAULT = "0"
        PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
        PROFILE_SELECTOR_PGE_PARAMETER = "N"
    END_OBJECT = PCF_ENTRY
    OBJECT = PCF_ENTRY
        CLASS = 181
        LOGICAL_ID = 424302
        PCF_FILE_TYPE = 5
        PGE_PARAMETER_NAME = "Processing Range Number of Lines"
        PGE_PARAMETER_DEFAULT = "0"
        PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
        PROFILE_SELECTOR_PGE_PARAMETER = "N"
    END_OBJECT = PCF_ENTRY
    OBJECT = PCF_ENTRY
        CLASS = 182
        LOGICAL_ID = 424303
        PCF_FILE_TYPE = 5
        PGE_PARAMETER_NAME = "Processing Range Begin Element"
        PGE_PARAMETER_DEFAULT = "0"
        PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
        PROFILE_SELECTOR_PGE_PARAMETER = "N"
    END_OBJECT = PCF_ENTRY
    OBJECT = PCF_ENTRY
        CLASS = 183
        LOGICAL_ID = 424304
        PCF_FILE_TYPE = 5
        PGE_PARAMETER_NAME = "Processing Range Number of Elements"
        PGE_PARAMETER_DEFAULT = "0"
        PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
        PROFILE_SELECTOR_PGE_PARAMETER = "N"
    END_OBJECT = PCF_ENTRY
END

```

A.4.8 GDAS_0ZF ODL

```

DATA_TYPE_NAME = "GDAS_0ZF"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "MODIS"
PLATFORM = "EOSAM1"
DATA_TYPE_DESCRIPTION = "NCEP 6-Hour Atmospheric Profile"
PROVIDER = "Goddard Space Flight Center"

```

```

NOMINAL_SIZE = 21.0
PROCESSING_LEVEL = "L1"
HDF_DATA = "N"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "NCEP"
PERIOD = "HOURS=6"
BOUNDARY = "START_OF_6HOUR"
DURATION = "SECS=1"
DELAY = 10
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
END

```

A.4.9 OZ_DAILY ODL

The same as GDAS_0ZF ODL except the following items:

```

DATA_TYPE_NAME = "OZ_DAILY"
DATA_TYPE_DESCRIPTION = "TOVS Column Ozone Daily Product"
NOMINAL_SIZE = 0.10
PERIOD = "DAYS=1"
BOUNDARY = "START_OF_DAY+43200"
DURATION = "SECS=1"

```

A.4.10 REYNSST ODL

The same as GDAS_0ZF ODL except the following items:

```

DATA_TYPE_NAME = "REYNSST"
DATA_TYPE_DESCRIPTION = "Reynolds Weekly SST"
NOMINAL_SIZE = 0.30
PERIOD = "SECS=604800"
BOUNDARY = "START_OF_WEEK-86400"
DURATION = "SECS=604800"

```

A.4.11 SEA_ICE ODL

The same as GDAS_0ZF ODL except the following items:

```

DATA_TYPE_NAME = "SEA_ICE"
DATA_TYPE_DESCRIPTION = "NCEP Ice Concentration"
NOMINAL_SIZE = 0.30
PERIOD = "SECS=86400"
BOUNDARY = "START_OF_DAY"

```

```
DURATION = "SECS=1"
```

A.4.12 NISE ODL

The same as GDAS_OZF ODL except the following items:

```
DATA_TYPE_NAME = "NISE"  
DATA_TYPE_DESCRIPTION = "NSIDC NISE snow/ice extent"  
NOMINAL_SIZE = 0.03  
PERIOD = "DAYS=1"  
BOUNDARY = "START_OF_DAY"  
DURATION = "DAYS=1"
```

A.5 Typical AIRS PGE & ESDT ODL Files

Listings are provided for the following AIRS ODL files:

A.5.1 AIRS PGE ODL for PGE_NAME AiL1A_AMSU

A.5.2 AIRS ESDT AIR10SCI ODL

A.5.3 AIRS ESDT AIR10SCC ODL

A.5.4 AIRS ESDT AIR20SCI ODL

A.5.5 AIRS ESDT PMCO_HK ODL

A.5.6 AIRS ESDT PM1EPHND ODL

A.5.7 AIRS ESDT PM1ATTNR ODL

A.5.8 AIRS ESDT AIRAASCI ODL

A.5.9 AIRS ESDT AIRXADCM ODL

A.5.10 AIRS ESDT AIRXATCM ODL

A.5.11 AIRS ESDT AIRXATCS ODL

A.5.12 AIRS ESDT AIRXARYL ODL

A.5.13 AIRS ESDT AIRXAGEO ODL

A typical AIRS PGE will differ from the examples here by the PGE_NAME, the specific input/output files referenced, and runtime parameters. However, the overall structure of a given AIRS PGE ODL file would be similar to the ones used here. (N.B. The ODL files shown here are associated with the AIRS version 2.1.2 software.)

A.5.1 AIRS PGE AiL1A_AMSU ODL

```
PGE_NAME = "L1A_AMSU"  
PGE_VERSION = "212"  
PROFILE_ID = 1
```

```

PROFILE_DESCRIPTION = "GRAN01"
PLATFORM = "EOSPM1"
INSTRUMENT = "AIRS"
MINIMUM_OUTPUTS = 0
SCHEDULE_TYPE = "Time"
PROCESSING_PERIOD = "MINS=6"
PROCESSING_BOUNDARY = "START_OF_DAY-31"
PGE_SSW_VERSION = "212"

/***** Primary Inputs *****/
OBJECT = PCF_ENTRY
  CLASS = 11
  LOGICAL_ID = 261
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "AIR10SCC"
  DATA_TYPE_VERSION = "001"
  BEGIN_PERIOD_OFFSET = 31
  END_PERIOD_OFFSET = 31
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 1
  INPUT_TYPE = "Required"
/*  ALIGN_DPR_TIME_WITH_INPUT_TIME = "Y"  */
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
  CLASS = 12
  LOGICAL_ID = 262
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "AIR10SCI"
  DATA_TYPE_VERSION = "001"
  BEGIN_PERIOD_OFFSET = 31
  END_PERIOD_OFFSET = 31
  MIN_GRANULES_REQUIRED = 1
  MAX_GRANULES_REQUIRED = 1
  INPUT_TYPE = "Required"
/*  ALIGN_DPR_TIME_WITH_INPUT_TIME = "Y"  */
  NUMBER_NEEDED = 1
  QUERY_TYPE = "Temporal"
  OBJECT = FILETYPE
    FILETYPE_NAME = "L0 Data Files"
    CLASS = 1
  END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
  CLASS = 13
  LOGICAL_ID = 290
  PCF_FILE_TYPE = 1
  DATA_TYPE_NAME = "AIR20SCI"
  DATA_TYPE_VERSION = "001"
  BEGIN_PERIOD_OFFSET = 31
  END_PERIOD_OFFSET = 31

```

```

MIN_GRANULES_REQUIRED = 1
MAX_GRANULES_REQUIRED = 1
INPUT_TYPE = "Required"
/* ALIGN DPR TIME WITH INPUT_TIME = "Y" */
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
OBJECT = FILETYPE
    FILETYPE_NAME = "L0 Data Files"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*****Dynamic ancillary inputs *****/
OBJECT = PCF_ENTRY
    CLASS = 14
    LOGICAL_ID = 4007
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "PMCO_HK"
    DATA_TYPE_VERSION = "001"
    BEGIN_PERIOD_OFFSET = 31
    END_PERIOD_OFFSET = 31
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    INPUT_TYPE = "Required"
    /* ALIGN DPR TIME WITH INPUT_TIME = "Y" */
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 14
    LOGICAL_ID = 4008
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "PMCO_HK"
    DATA_TYPE_VERSION = "001"
    BEGIN_PERIOD_OFFSET = 31
    END_PERIOD_OFFSET = 31
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    INPUT_TYPE = "Required"
    /* ALIGN DPR TIME WITH INPUT_TIME = "Y" */
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/**** Attitude/Ephemeris/DEM entry. Please delete if not used by PGE. ****/
OBJECT = PCF_ENTRY
    CLASS = 18
    LOGICAL_ID = 10501
    PCF_FILE_TYPE = 1

```

```

DATA_TYPE_NAME = "PM1EPHND"
DATA_TYPE_VERSION = "001"
BEGIN_PERIOD_OFFSET = 31
END_PERIOD_OFFSET = 31
MIN_GRANULES_REQUIRED = 1
MAX_GRANULES_REQUIRED = 1
INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 19
    LOGICAL_ID = 10502
    PCF_FILE_TYPE = 1
    DATA_TYPE_NAME = "PM1ATTNR"
    DATA_TYPE_VERSION = "001"
    BEGIN_PERIOD_OFFSET = 31
    END_PERIOD_OFFSET = 31
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*****Primary output *****/
OBJECT = PCF_ENTRY
    CLASS = 110
    LOGICAL_ID = 7120
    PCF_FILE_TYPE = 2
    DATA_TYPE_NAME = "AIRAASCI"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULE_YIELD = 1
    MAX_GRANULE_YIELD = 1
    ASSOCIATED_MCF_ID = 17120
    SCIENCE_GROUP = "S1"
    MINIMUM_SIZE = 1
    MAXIMUM_SIZE = 100
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

/*****Static ancillary inputs *****/
OBJECT = PCF_ENTRY
    CLASS = 116
    LOGICAL_ID = 4001
    PCF_FILE_TYPE = 3

```



```

DATA_TYPE_NAME = "AIRXADCM"
DATA_TYPE_VERSION = "001"
MIN_GRANULES_REQUIRED = 1
MAX_GRANULES_REQUIRED = 1
SCIENCE_GROUP = "0001"
INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 117
    LOGICAL_ID = 4002
    PCF_FILE_TYPE = 3
    DATA_TYPE_NAME = "AIRXATCM"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "0002"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 118
    LOGICAL_ID = 4003
    PCF_FILE_TYPE = 3
    DATA_TYPE_NAME = "AIRXATCS"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "0003"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 119
    LOGICAL_ID = 4005
    PCF_FILE_TYPE = 3
    DATA_TYPE_NAME = "AIRXARYL"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1

```

```

SCIENCE_GROUP = "O004"
INPUT_TYPE = "Required"
NUMBER_NEEDED = 1
QUERY_TYPE = "Temporal"
OBJECT = FILETYPE
    FILETYPE_NAME = "Single File Granule"
    CLASS = 1
END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 120
    LOGICAL_ID = 4006
    PCF_FILE_TYPE = 3
    DATA_TYPE_NAME = "AIRXAGEO"
    DATA_TYPE_VERSION = "001"
    MIN_GRANULES_REQUIRED = 1
    MAX_GRANULES_REQUIRED = 1
    SCIENCE_GROUP = "O005"
    INPUT_TYPE = "Required"
    NUMBER_NEEDED = 1
    QUERY_TYPE = "Temporal"
    OBJECT = FILETYPE
        FILETYPE_NAME = "Single File Granule"
        CLASS = 1
    END_OBJECT = FILETYPE
END_OBJECT = PCF_ENTRY

OBJECT = PCF_ENTRY
    CLASS = 125
    LOGICAL_ID = 1001
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Instrument: 0=AMSU, 1=AIRS, 2=HSB(MHS), 3=VIS"
    PGE_PARAMETER_DEFAULT = "0"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 225
    LOGICAL_ID = 1002
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Print Level IO: 0=Off, 1=Low, 2=Med, 3=High"
    PGE_PARAMETER_DEFAULT = "2"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 226
    LOGICAL_ID = 1003
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Print Level: 0=Off, 1=Low, 2=Med, 3=High"
    PGE_PARAMETER_DEFAULT = "1"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 227
    LOGICAL_ID = 1004
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Exec Development Mode: 0=Off, 1=On"
    PGE_PARAMETER_DEFAULT = "0"

```

```

    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 228
    LOGICAL_ID = 1005
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Run Level 2 Mode: 1=MIT & 2=NOAA & 4=GSFC"
    PGE_PARAMETER_DEFAULT = "7"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 126
    LOGICAL_ID = 1006
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Stats Mode: 0=Off, 1=cmp2truth, 2=cmp2MW-retrieval"
    PGE_PARAMETER_DEFAULT = "0"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 128
    LOGICAL_ID = 1011
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Year (ex: 1998)"
    PGE_PARAMETER_DEFAULT = "1998"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 129
    LOGICAL_ID = 1012
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Month number (1 - 12)"
    PGE_PARAMETER_DEFAULT = "09"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 130
    LOGICAL_ID = 1013
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Day of month (1 - 31)"
    PGE_PARAMETER_DEFAULT = "13"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 131
    LOGICAL_ID = 1014
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Orbit of day (1 - 17)"
    PGE_PARAMETER_DEFAULT = "1"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
    CLASS = 132
    LOGICAL_ID = 1015
    PCF_FILE_TYPE = 5
    PGE_PARAMETER_NAME = "Granule Number (1 - 17)"
    PGE_PARAMETER_DEFAULT = "01"
    PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY

```

```

OBJECT = PCF_ENTRY
  CLASS = 133
  LOGICAL_ID = 1016
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Granule Size in scansets (1 - 45)"
  PGE_PARAMETER_DEFAULT = "45"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 134
  LOGICAL_ID = 1020
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Times Processed: 1 for never before reprocessed"
  PGE_PARAMETER_DEFAULT = "1"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 135
  LOGICAL_ID = 1021
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "Processing Facility: A for TLSCF or G for GDAAC"
  PGE_PARAMETER_DEFAULT = "G"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
OBJECT = PCF_ENTRY
  CLASS = 200
  LOGICAL_ID = 411
  PCF_FILE_TYPE = 5
  PGE_PARAMETER_NAME = "GDAAC Build Version String"
  PGE_PARAMETER_DEFAULT = "PGE=2.1.2, SDPTK=5.2.7.2, HDF=4.1r3,HDFEOS=2.7,
OS=6.5, COMPILER=7.2.1.3, ECS=6A.03"
  PGE_PARAMETER_DYNAMIC_VALUE = "NONE"
END_OBJECT = PCF_ENTRY
END

```

A.5.2 AIRS ESDT AIR10SCI ODL

```

DATA_TYPE_NAME = "AIR10SCI"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "AIRS"
PLATFORM = "EOSPM1"
DATA_TYPE_DESCRIPTION = "AMSU_A1 Science Data Packets"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = .02
PROCESSING_LEVEL = "L0"
HDF_DATA = "N"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "GSFC"
PERIOD = "HOURS=2"
BOUNDARY = "START_OF_DAY"
SPATIAL_FLAG = "N"
DELAY = 43200
OBJECT = USE_OBJECT
  CLASS = 1
  USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"

```

```

PROCESSED_AT = "GSFC"
OBJECT = FILETYPE
  CLASS = 1
  FILETYPE_NAME = "L0 Data Files"
  MAXIMUM_NUM_FILES = 2
END_OBJECT = FILETYPE
END

```

A.5.3 AIRS ESDT AIR10SCC ODL

```

DATA_TYPE_NAME = "AIR10SCC"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "AIRS"
PLATFORM = "EOSPM1"
DATA_TYPE_DESCRIPTION = "AMSU_A1 Science Data Packets"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = .02
PROCESSING_LEVEL = "L0"
HDF_DATA = "N"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "GSFC"
PERIOD = "HOURS=2"
BOUNDARY = "START_OF_DAY"
SPATIAL_FLAG = "N"
DELAY = 43200
OBJECT = USE_OBJECT
  CLASS = 1
  USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
OBJECT = FILETYPE
  CLASS = 1
  FILETYPE_NAME = "L0 Data Files"
  MAXIMUM_NUM_FILES = 2
END_OBJECT = FILETYPE
END

```

A.5.4 AIRS ESDT AIR20SCI ODL

```

DATA_TYPE_NAME = "AIR20SCI"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "AIRS"
PLATFORM = "EOSPM1"
DATA_TYPE_DESCRIPTION = "AMSU_A2 Science Data Packets"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = .02
PROCESSING_LEVEL = "L0"
HDF_DATA = "N"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "GSFC"
PERIOD = "HOURS=2"
BOUNDARY = "START_OF_DAY"
SPATIAL_FLAG = "N"
DELAY = 43200
OBJECT = USE_OBJECT

```

```

        CLASS = 1
        USED_BY = "GSFC"
    END_OBJECT = USE_OBJECT
    ARCHIVED_AT = "GSFC"
    PROCESSED_AT = "GSFC"
    OBJECT = FILETYPE
        CLASS = 1
        FILETYPE_NAME = "L0 Data Files"
        MAXIMUM_NUM_FILES = 2
    END_OBJECT = FILETYPE
END

```

A.5.5 AIRS ESDT PMCO_HK ODL

```

DATA_TYPE_NAME = "PMCO_HK"
DATA_TYPE_VERSION = "001"
DATA_TYPE_DESCRIPTION = "Aqua Carryout housekeeping file"
INSTRUMENT = "All"
PLATFORM = "Aqua"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 2.0
PROCESSING_LEVEL = "L0"
DYNAMIC_FLAG = "E"
PREDICTION_METHOD = "ROUTINE"
SUPPLIER_NAME = "GSFC"
PERIOD = "HOURS=2"
BOUNDARY = "START_OF_DAY"
DELAY = 43200
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
HDF_DATA = "N"
END

```

A.5.6 AIRS ESDT PM1EPHND ODL

```

DATA_TYPE_NAME = "PM1EPHND"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "All"
PLATFORM = "PM1"
DATA_TYPE_DESCRIPTION = "PM-1 FDD Definitive Ephemeris data in Toolkit format"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 6.0
PROCESSING_LEVEL = "L1"
DYNAMIC_FLAG = "I"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
HDF_DATA = "N"
END

```

A.5.7 AIRS ESDT PM1ATTNR ODL

```
DATA_TYPE_NAME = "PM1ATTNR"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "All"
PLATFORM = "PM1"
DATA_TYPE_DESCRIPTION = "PM-1 Refined Attitude Data in Toolkit format"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 1.0
PROCESSING_LEVEL = "L1"
DYNAMIC_FLAG = "I"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
HDF_DATA = "N"
END
```

A.5.8 AIRS ESDT AIRAASCI ODL

```
DATA_TYPE_NAME = "AIRAASCI"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "AIRS"
PLATFORM = "EOSPM1"
DATA_TYPE_DESCRIPTION = "AMSU-A geolocated science counts"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 3.0
PROCESSING_LEVEL = "L1A"
HDF_DATA = "N"
PERIOD = "SECS=360"
BOUNDARY = "START_OF_SEC"
DYNAMIC_FLAG = "I"
SPATIAL_FLAG = "N"
OBJECT = USE_OBJECT
    CLASS = 1
    USED_BY = "GSFC"
END_OBJECT = USE_OBJECT
ARCHIVED_AT = "GSFC"
PROCESSED_AT = "GSFC"
END
```

A.5.9 AIRS ESDT AIRXADCM ODL

```
DATA_TYPE_NAME = "AIRXADCM"
DATA_TYPE_VERSION = "001"
INSTRUMENT = "AIRS"
PLATFORM = "EOSPM1"
DATA_TYPE_DESCRIPTION = "Decommuration map"
PROVIDER = "Goddard Space Flight Center"
NOMINAL_SIZE = 1.0
PROCESSING_LEVEL = "L1A"
HDF_DATA = "N"
DYNAMIC_FLAG = "S"
SPATIAL_FLAG = "N"
END
```

A.5.10 AIRS ESDT AIRXATCM ODL

```
DATA_TYPE_NAME = "AIRXATCM"  
DATA_TYPE_VERSION = "001"  
INSTRUMENT = "AIRS"  
PLATFORM = "EOSPM1"  
DATA_TYPE_DESCRIPTION = "Conversion method file"  
PROVIDER = "Goddard Space Flight Center"  
NOMINAL_SIZE = 1.0  
PROCESSING_LEVEL = "L1A"  
HDF_DATA = "N"  
DYNAMIC_FLAG = "S"  
SPATIAL_FLAG = "N"  
END
```

A.5.11 AIRS ESDT AIRXATCS ODL

```
DATA_TYPE_NAME = "AIRXATCS"  
DATA_TYPE_VERSION = "001"  
INSTRUMENT = "AIRS"  
PLATFORM = "EOSPM1"  
DATA_TYPE_DESCRIPTION = "Constant sets"  
PROVIDER = "Goddard Space Flight Center"  
NOMINAL_SIZE = 1.0  
PROCESSING_LEVEL = "L1A"  
HDF_DATA = "N"  
DYNAMIC_FLAG = "S"  
SPATIAL_FLAG = "N"  
END
```

A.5.12 AIRS ESDT AIRXARYL ODL

```
DATA_TYPE_NAME = "AIRXARYL"  
DATA_TYPE_VERSION = "001"  
INSTRUMENT = "AIRS"  
PLATFORM = "EOSPM1"  
DATA_TYPE_DESCRIPTION = "Red Yellow limits"  
PROVIDER = "Goddard Space Flight Center"  
NOMINAL_SIZE = 1.0  
PROCESSING_LEVEL = "L1A"  
HDF_DATA = "N"  
DYNAMIC_FLAG = "S"  
SPATIAL_FLAG = "N"  
END
```

A.5.13 AIRS ESDT AIRXAGEO ODL

```
DATA_TYPE_NAME = "AIRXAGEO"  
DATA_TYPE_VERSION = "001"  
INSTRUMENT = "AIRS"  
PLATFORM = "EOSPM1"  
DATA_TYPE_DESCRIPTION = "L1A.geolocation.anc"  
PROVIDER = "Goddard Space Flight Center"  
NOMINAL_SIZE = 1.0  
PROCESSING_LEVEL = "L1A"  
HDF_DATA = "N"  
DYNAMIC_FLAG = "S"  
SPATIAL_FLAG = "N"  
END
```


A.6 Typical Aqua MODIS PGE ODL File

Listings are provided for the following MODIS ODL files:

A.6.1 ODL files for Aqua MODIS PGE

The Aqua PGE ODL files are similar to those of Terra PGEs. The main differences are DATA_TYPE_NAME for input and output granules:

	Terra	Aqua
-----	-----	-----
L0	MOD000	MODPML0
Static input	MOD01LUT	MYD01LUT
	MOD03LUT	MYD03LUT
	MOD02LUT	MYD02LUT
	MOD07LUT	MYD07LUT
	MOD35ANC	MYD35ANC
Ephemeris	AM1EPHN0	PM1EPHND
Attitude	AM1ATTNF	PM1ATTNR
Products	MOD01	MYD01
	MOD03	MYD03
	MOD02QKM	MYD02QKM
	MOD02HKM	MYD02HKM
	MOD021KM	MYD021KM
	MOD02OBC	MYD02OBC
	MODVOLC	MYDVOLC
	MOD07_L2	MYD07_L2
	MOD07_QC	MYD07_QC
	MOD35_L2	MYD35_L2
	MOD35_QC	MYD35_QC
	MODCSR_G	MYDCSR_G

This page intentionally left blank.

Abbreviations and Acronyms

A _o	Operational Availability
ACL	Access Control List
ADC	Affiliated Data Center (NOAA)
ADSERV	Advertising Service
AI&T	Algorithm Integration and Test
AIT	Algorithm Integration Team
AMASS	Archival Management and Storage System
ASCII	American Standard Code for Information Interchange
BBS	Bulletin Board System
CCB	Configuration Control Board (NASA Convention)
CCR	Configuration Change Request
CCRS	Canada Centre for Remote Sensing
CD	Compact Disk
CDE	Common Desktop Environment
CD-ROM	Compact Disk - Read Only Memory
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CDS	Cell Directory Service
CHCI	Communications Hardware Configuration Item
CHUI	Character User Interface
CI	Configuration Item
CIDM	Client Interoperability and Data Management
CM	Configuration Management
CMA	CM Administrator
CN	Change Notice
CO	Contracting Officer
COTR	Contracting Officer's Technical Representative

COTS	Commercial Off-the-Shelf (hardware or software)
CPU	Central Processing Unit
CR	Change Request
CRM	Change Request Manager
CSCI	Computer Software Configuration Item
CSMS	Communications and Systems Management Segment (ECS)
CSR	Consent To Ship Review
CSS	Communication Subsystem
DAA	Data Availability Acknowledgment
DAAC	Distributed Active Archive Center
DADS	Data Archive and Distribution System
DAN	Data Availability Notice
DAP	Delivered Algorithm Package
DAR	Data Acquisition Request
DAS	Data Availability Schedule
DAT	Digital Audio Tape
DB	Database
DBA	Database Administrator
DBMS	Database Management System
DCE	Distributed Computing Environment (OSF)
DCF	Data Capture Facility
DCN	Document Change Notice
DCO	Document Change Order
DCR	Data Collection Request
DD	Data Dictionary
DDA	Data Delivery Acknowledgment
DDICT	Data Dictionary
DDIST	Data Distribution
DDN	Data Delivery Notice
DDSRV	Document Data Server

DDTS	Distributed Defect Tracking System
DES	Data Encryption Standard
DESKT	Desktop Configuration Item
DID	Data Item Description
DIF	Data Interchange Formant
DIMGR	Distributed Information Manager
DME	Distributed Management Environment
DMO	Data Management Organization
DNS	Domain Name Service
DOF	Distributed Object Framework
DP	Data Pool
DPR	Data Processing Request
DPS	Data Processing Subsystem
DR	Delivery Record
DS	Data Server
DSS	Data Server Subsystem
e-mail	Electronic Mail
EBnet	EOSDIS Backbone Network (combines Ecom and ESN)
ECN	Equipment Control Number
ECS	EOSDIS Core System
EDC	EROS Data Center (DAAC)
EDF	ECS Development Facility
EDHS	ECS Data Handling System
EGS	EOS Ground System
EIN	Equipment Identification Number
EMC	Enterprise Monitoring and Coordination
EMSn	EOS Mission Support Network (formerly EBnet)
EOC	EOS Operations Center (ECS)
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System

EP	Evaluation Package
EROS	Earth Resources Observation System
ESD	Electrostatic Discharge
ESDIS	Earth Science Data and Information System (GSFC Code 505)
ESDT	Earth Science Data Type
ESOD	Earth Science On-line Directory
ET	Eastern (standard or daylight savings) Time
FDDI	Fiber Distributed Data Interface
FORTRAN	FORmula TRANslation (computer language)
FOS	Flight Operations Segment (ECS)
FOT	Flight Operations Team
FTP	File Transfer Protocol
FTPD	File Transfer Protocol Daemon
GB	Gigabyte (10^9)
Gb	Gigabit (10^9)
GBps	Gigabytes per Second
Gbps	Gigabits per Second
GCDIS	Global Change Data and Information System
GCMD	Global Change Master Directory
GFE	Government Furnished Equipment
GFP	Government Furnished Property
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
H/W	Hardware
HDF	Hierarchical Data Format
HIPPI	High Performance Parallel Interface
HPOV	HP Open View
HSM	Hierarchical Storage Management
HTML	Hypertext Mark-Up Language
HWCi	Hardware Configuration Item

I&AT	Integration and Acceptance Test
I&T	Integration and Test
I&TT	Integration and Test Team
IATO	Independent Acceptance Test Organization
ICD	Interface Control Document
ICLHW	Ingest Client Hardware [configuration item]
ILM	Inventory, Logistics, and Maintenance
ILP	Integrated Logistics Plan
ILS	Integrated Logistics Support
ILSMT	ILS Management Team
ILSO	ILS Office
INGST	Ingest Services
INS	Ingest System
IOS	Interoperability Subsystem
IP	Internet Protocol
IQ	Intelligent Query and IQ Access
Ir1	Interim Release 1
ISDN	Integrated Services Digital Network
ISS	Internetworking Subsystem
ISQL	Interactive SQL
IV&V	Independent Verification and Validation
JIL	Job Information Language
JPL	Jet Propulsion Laboratory (DAAC)
KB	Kilobyte (10^3)
Kb	Kilobit (10^3)
KBps	Kilobytes per Second
Kbps	Kilobits per Second
L-7	Landsat-7 (Landsat-7 for EDHS search)
L0	Level 0
L0-L4	Level 0 (zero) through Level 4 (use Level-0 through Level-4 for EDHS search)

LAN	Local Area Network
Landsat	Land Remote-Sensing Satellite
LaRC	Langley Research Center (DAAC)
LDOS	Landsat Data and Operations System
LIM	Local Information Manager
LIMGR	Local Information Manager
LMC	Local Maintenance y
Loral	Loral Aerosys (ECS Team)
LRU	Line Replaceable Unit
M	Million, mega (prefix)
M&O	Maintenance and Operations
MAN	Metropolitan Area Network
MB	Megabyte (10^6)
Mb	Megabit (10^6)
MBps	Megabytes per Second
Mbps	Megabits per Second
MCF	Metadata Configuration File Metadata Control File
MD	Master Directory
MDA	Management Data Access
MDT	Mean Downtime
MHWCI	Management Hardware Configuration Item
MHz	Megahertz
MIB	Management Information Base
MIS	Management Information System
MM	Millimeter
MO&DSD	Mission Operations and Data Systems Directorate (GSFC Code 500)
MOU	Memorandum of Understanding
MR	Malfunction Report
MSEC	Millisecond

MSFC	Marshall Space Flight Center (DAAC)
MSS	Management Systems Subsystem
MTBCM	Mean Time Between Corrective Maintenance
MTBF	Mean Time Between Failure
MTBM	Mean Time Between Maintenance
MTBPM	Mean Time Between Preventive Maintenance
MTPE	Mission to Planet Earth
MTTR	Mean Time to Repair
MTTRes	Mean Time to Restore
N/A	Not Applicable
NA	Network Administrator
NASA	National Aeronautics and Space Administration
Nascom	NASA Communications
NCC	Network Control Center (GSFC) network communication center
NCR	Nonconformance Report
NCS	Netscape Commerce Server
NCSA	National Center for Supercomputer Applications
NMCI	Network Management Configuration Item
NNM	HPOpenView Network Node Manager
NOAA	National Oceanic and Atmospheric Administration
NSI	NASA Science Internet
NWCI	Networking Configuration Item
OEM	Original Equipment Manufacturer
OJT	On-the-Job Training
OPS	Operations
Ops Super	Operations Supervisor
ORPA	Operations Readiness & Performance Assurance
ORR	Operations Readiness Review
OS	Operating System

OSF	Open Software Foundation
OTS	Off the Shelf
OVW	HP OpenView Windows
PAIP	Performance Assurance Implementation Plan
PB	Petabyte (10^{15})
PC	Personal Computer Process Control
PCF	Process Control File
PDL	Program Design Language
PDPS	Planning and Data Processing System
PDS	Product Distribution System
PDSOI	Product Distribution System Operator Interface
PDSIS	Product Distribution System Interface Server
PGE	Product Generation Executive
PGS	Product Generation Service
PI	Principal Investigator
PIN	Password Identification Number
PLANG	Production Planning CSCI
PLNHW	Planning Hardware [configuration item]
PLS	Planning Subsystem
PM	Preventative Maintenance
PPM	Principal Period of Maintenance
PR	Production Request
QA	Quality Assurance
QC	Quality Control
QRU	Query, Retrieve, and Update
R&M	Reliability and Maintainability
RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RE	Responsible Engineer

RID	Review Item Discrepancy
RMA	Return Material Authorization
RMS	Resource Management Subsystem
RSM	Replication Server Manager
RSSD	Replication Server System Database
S/C	Spacecraft
S/W	Software
S/WCI	Software Configuration Item
SA	System Administrator
SATAN	Security Administrator Tool for Analyzing Networks
SCDO	Science and Communications Development Office (ECS)
SCF	Science Computing Facility
SCSI	Small Computer System Interface
SDP	Science Data Processing
SDPF	Science Data Processing Facility
SDPS	Science Data Processing Segment (ECS)
SDPS/W	Science Data Production Software
SDPTK	Science Data Processing Toolkit
SDSRV	Science Data Server
SE	System Engineering
SE&I	System Engineering and Integration
SEI&T	System Engineering, Integration, and Test
SEO	Sustaining Engineering Organization
SEPG	Software Engineering Process Group
SGI	Silicon Graphics Incorporated
SI&T	System Integration and Test
SMC	System(s) Monitoring and Coordination Center
SMF	Status Message Facility
SMTP	Simple Mail Transport Protocol
SNMP	Simple Network Management Protocol

SOR	System Operations Review
SORR	Segment Operational Readiness Review
SPRHW	Science Processing Hardware [configuration item]
SQL	Structured Query Language
SQR	SQL Report Writer
SRR	System Requirements Review
SSAP	Science Software Archive Package
SSI&T	Science Software Integration and Test
SSL	Secure Socket Layer
STMGT	Storage Management
T&M	Time and Materials
TB	Terabyte (10^{12})
TBC	To Be Confirmed
TBD	To Be Determined
TBR	To Be Resolved
TBS	To Be Supplied
Tbyte	Terabyte
TCP/IP	Transmission Control Protocol/Internet Protocol
TEC	Tivoli Enterprise Console
telecon	Telephone Conference
TELNET	Telecommunication Network
TRMM	Tropical Rainfall Measurement Mission
TSDIS	TRMM Science Data and Information System
TT	Trouble Ticket
UDP	User Datagram Protocol
UR	Universal Reference
URDB	User Recommendations Database
URL	Universal Resource Locator
USO	User Support Office
US Rep	User Services Representative

UWG	User Working Group
VDD	Version Description Document
VOB	Versioned Object Base (ClearCase)
WAIS	Wide Area Information Server
WAN	Wide Area Network
WKBCH	Workbench
WKSHCI	Working Storage Hardware Configuration Item
WWW	World Wide Web

This page intentionally left blank.